PENGENDALI NOISE AKTIF DENGAN FUZZY LOGIC

TUGAS AKHIR

Oleh :

ERWIN DIAN SANTOSO
NRP. 2291.100.038

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2000

PENGENDALI NOISE AKTIF DENGAN FUZZY LOGIC

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Pada
Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui / Menyetujui

Dosen Pembimbing

Ir. SOETIKNO
NIP. 130 445 231

SURABAYA
Pebruari, 2000

ABSTRAK

Fuzzy logic dapat digunakan dalam proses penghilangan noise, yaitu sebagai pengontrol dari sistem yang menghasilkan sinyal anti noise yang frekuensi dan amplitudonya sama dengan sinyal noise tetapi fasenya berlawanan. Sehingga hasil superposisi antara sinyal noise dan sinyal anti noise ialah sinyal yang amplitudonya mendekati nol karena kedua sinyal saling melemahkan.

Sinyal anti noise dihasilkan dari sinyal noise yang frekuensinya tertentu setelah mengalami pergeseran fase dan penyesuaian amplitudo. Hasil interferensi antara sinyal noise dan sinyal anti noise menghasilkan sinyal error yang merupakan sinyal umpan balik ke sistem.

Dalam tugas akhir ini digunakan dua buah *Fuzzy logic controller* AL220. *Fuzzy logic controller* yang pertama digunakan untuk menentukan nilai RMS dari sinyal error yang ada. *Fuzzy logic controller* yang kedua digunakan untuk menentukan besarnya faktor penguatan dan besarnya pergeseran fase yang diberikan terhadap sinyal input yang berupa sinyal noise. *Fuzzy logic controller* ini bertugas mengontrol dua buah operasional transkonduktansi amplifier LM13600 yang masing-masing berfungsi sebagai rangkaian penggeser fase dan rangkaian pengatur besarnya faktor penguatan sehingga dihasilkan sinyal anti noise yang sesuai untuk menghilangkan sinyal noise yang ada. Sebagai sumber noise digunakan kipas angin yang menghasilkan sinyal noise dengan frekuensi 120 Hz dan sebagai sumber sinyal anti noise digunakan speaker.

KATA PENGANTAR

Segala puji dan syukur penyusun panjatkan kepada Tuhan Yang Mahaesa karena berkat dan anugrah-Nya Tugas Akhir ini dapat diselesaikan.

Tugas Akhir ini merupakan salah satu persyaratan untuk memperoleh gelar Sarjana Teknik pada Bidang Studi Elektronika, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institus Sepuluh Nopember Surabaya.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna.

Dengan telah selesainya penyusunan Tugas Akhir ini, maka penyusun mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Soetikno selaku dosen pembimbing I dan koordinator Bidang Studi Elektronika yang telah memberikan saran-saran serta dorongan.
2. Bapak Dr.Ir.Ahmad Jazidie selaku ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.
3. Lydia yang selalu menemani, membantu dan memberi semangat.
4. Rekan-rekan yang telah membantu dalam pembuatan tugas akhir ini.

Sebagai akhir kata penyusun berharap semoga tugas akhir ini dapat bermanfaat bagi para pembaca

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

Bab ini berupa pendahuluan dari bab-bab yang ada. Pada bab ini akan dijelaskan tentang latar belakang penelitian, permasalahan yang ada, tujuan penelitian, pembatasan terhadap permasalahan yang ada, metodologi yang digunakan dalam penelitian, sistematika penulisan, dan relevansi dari penelitian yang dilakukan.

1.1. LATAR BELAKANG

Masalah utama yang sering terjadi dalam suatu sistem ialah timbulnya noise. Agar tidak menimbulkan hal-hal yang mengganggu atau hal-hal yang merugikan sedapat mungkin noise yang timbul dihilangkan.

Suatu sinyal bila ditambahkan dengan sinyal yang sama tapi fasenya berlawanan akan saling meniadakan. Bila kita mampu menghasilkan suatu sinyal yang frekuensi dan amplitudonya sama dengan sinyal noise yang ada tetapi fasenya berlawanan, maka hasil penjumlahan kedua sinyal tersebut nol dan noise yang ada dapat dihilangkan.

1.2. PERMASALAHAN

Permasalahan yang dihadapi dalam Tugas Akhir ini, diarahkan pada masalah mengenai:

- Proses pencuplikan sinyal noise yang ada.
- Pemanfaatan logika fuzzy untuk menentukan amplitudo dan fase dari sinyal anti noise.
- Proses pembentukan sinyal anti noise berdasarkan parameter yang dihasilkan oleh logika fuzzy.

1.3. TUJUAN

Tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut :

- Mempelajari teori pencuplikan sinyal.
- Mempelajari teori Fuzzy Logic dan penerapannya.
- Merancang dan membuat suatu sistem yang menghasilkan sinyal anti noise.

1.4. BATASAN PERMASALAHAN

Sinyal anti noise dapat dihasilkan dengan tepat bila frekuensi sinyal noise tertentu. Parameter yang diatur dalam sinyal anti noise ialah amplitudo dan fasenya.

Dalam tugas akhir ini sinyal noise berasal dari udara yang bergerak yang dihasilkan oleh suatu kipas angin. Sensor noise ini berupa sebuah mikrofon kecil yang dilekatkan di rumah kipas angin yang digunakan.

1.5. METODOLOGI

Untuk merealisasi tujuan yang ingin dicapai Tugas Akhir ini, maka langkah-langkah yang perlu dilakukan adalah :

- Studi literatur (kepustakaan).
- Perancangan dan pembuatan sistem, baik perangkat keras maupun perangkat lunak.
- Uji coba sistem
- Penyusunan buku laporan tugas akhir (skripsi).

1.6. SISTEMATIKA

Buku laporan tugas akhir ini disusun dengan sistematika sebagai berikut :

- Bab I merupakan bab pendahuluan, yang menjelaskan tentang : Latar Belakang; Permasalahan; Tujuan; Batasan Permasalahan; Metodologi; Sistematika; dan Relevansi dari pembuatan tugas akhir ini.
- Bab II merupakan bab penjelasan dari teori-teori yang menunjang tugas akhir.
- Bab III menjelaskan tentang perancangan, yang berupa perangkat keras maupun perangkat lunak, dan pembuatan sistem.
- Bab IV meliputi uji coba dari sistem.
- Bab V merupakan bab penutup, yang terdiri dari kesimpulan dan saran.

1.7. RELEVANSI

Diharapkan studi yang dilakukan dalam tugas akhir ini dapat menghasilkan suatu pengetahuan tentang penggunaan fuzzy logic dalam proses pembentukan sinyal anti noise yang mana proses tersebut merupakan proses yang

kompleks. Logika fuzzy dapat digunakan dalam suatu sistem yang nonlinier dan dinamis.

BAB II

TEORI PENUNJANG

Pada bab ini akan dijelaskan tentang teori dasar yang menunjang penelitian yang dilakukan. Teori dasar tersebut meliputi teori tentang rangkaian konversi analog ke digital, teori tentang logika fuzzy, teori tentang mikrokontroler *fuzzy logic* yang digunakan yaitu AL220, teori filter analog, dan teori tentang penguat transkonduktansi. Pembahasan tentang mikrokontroler fuzzy AL220 meliputi blok diagram AL220, fungsi dari pin-pin yang ada, variabel fuzzy yang digunakan, *membership function*, *rule*, *floating membership function*, *fuzzifier*, pembaharuan *output latch*, metode defuzzifikasi, organisasi memori, dan sistem pewaktuan AL220. Sedangkan pembahasan tentang filter analog meliputi jenis-jenis filter analog beserta rangkaiannya. Pembahasan tentang transkonduktansi amplifier meliputi rangkaian penguat terkontrol tegangan dan rangkaian tahanan terkontrol tegangan.

2.1.RANGKAIAN KONVERSI ANALOG DIGITAL (ADC)

ADC digunakan untuk mengubah tegangan analog menjadi bentuk digital dalam bentuk bit-bit sehingga dapat diproses secara digital. Pengubah analog ke digital ini berfungsi untuk mencuplik sinyal masukan.

Beberapa faktor yang perlu diperhatikan dalam pemilihan komponen ADC antara lain

1. Resolusi

   Resolusi merupakan Specifikasi terpenting dari ADC, yaitu kemampuan perubahan terkecil dari tegangan masukkan untuk mengubah kode biner yang sekarang menjadi kode biner berikutnya pada keluaran ADC.

2. Akurasi

   Akurasi adalah jumlah dari semua kesalahan, misalnya kesalahan non-linearitas, skala penuh, skala nol dan lain-lain. Akurasi dapat juga menyatakan perbedaan antara tegangan masukkan analog secara teoritis terhadap masukkan yang nyata yang menghasilkan tegangan kode tersebut.

3. Waktu konversi

   Waktu yang dibutuhkan untuk mengubah besaran analog menjadi bentuk digital untuk setiap samplenya atau waktu yang diperlukan untuk melakukkan satu konversi.

4. Daerah tegangan input

   Daerah tegangan input ADC menentukan range tegangan sinyal input yang dapat dikonversi. Daerah tegangan ini bergantung pada tegangan referensi yang diberikan pada ADC.

2.2. MIKROKONTROLER *FUZZY LOGIC*

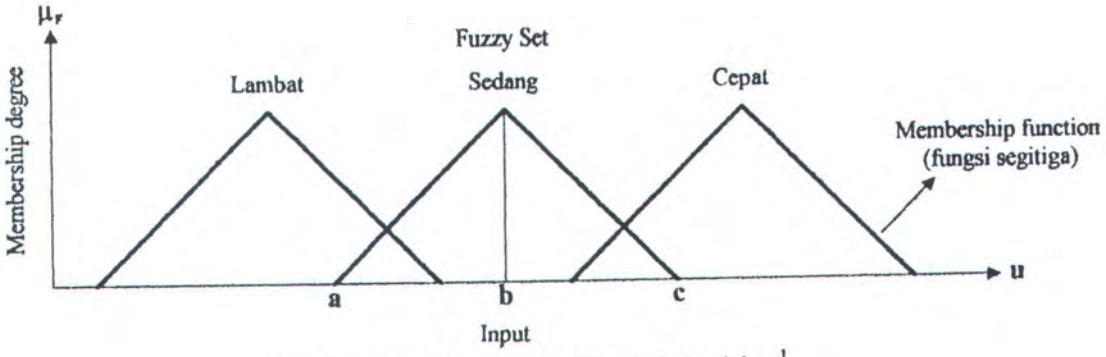
2.2.1. PENDAHULUAN

Teori *Fuzzy Logic* mula-mula dirumuskan oleh Lotfi Zadeh pada tahun 1965. Kegunaan utama *Fuzzy Logic* dalam sistem kontrol adalah untuk

memecahkan fungsi yang dinamis dan non-linier dengan *term* dan *rule* yang dapat dimengerti.

Pendekatan fuzzy lebih baik dibandingkan pendekatan konvensional. Perbedaan tersebut terletak pada kontrol logika fuzzy, term dan rule-nya, tidak dievaluasi hanya pada dua nilai "Benar" atau "Salah", tetapi mempunyai derajat keanggotaan pada multi nilai. Rule fuzzy sendiri mempunyai range kemungkinan yang berkelanjutan.

Pertimbangan yang digunakan dalam mengambil keputusan terdiri dari pertimbangan kualitatif dan kuantitatif. Pertimbangan kuantitatif biasanya diambil berdasarkan numerik, sedangkan kualitatif bersifat tidak numerik, yaitu *IF-THEN*. Teori Fuzzy set merupakan sarana matematis yang sangat baik dalam menyatakan hukum kualitatif *IF-THEN* tersebut, sehingga dapat diimplementasikan pada komputer. Pada konsep Fuzzy set, suatu obyek diberi nilai atau harga (*grade*) atas keberadaannya (*fuzzy membership function*), yang menyatakan tingkat kedekatan dengan titik referensi dari set. Fuzzy membership function ini umumnya berbentuk fungsi S, fungsi π, fungsi segitiga, fungsi trapezoidal, dan fungsi eksponensial. Notasi matematis fuzzy set ditulis sebagai berikut : $F = \{(u, \mu_F(u)) \mid u \varepsilon U\}$. Adapun notasi membership degree μ_F: U(0,1). Contoh dari fuzzy set, fungsi keanggotaan, dan derajat keanggotaan (*membership degree*) fungsi segitiga dapat dilihat pada gambar berikut,



Gambar 2.1. Fuzzy Sets Fungsi Segitiga[1].

2.2.2. OPERASI HIMPUNAN FUZZY (*FUZZY SET*)

Terdapat empat operasi dasar yang dilakukan untuk perhitungan pada himpunan fuzzy berdasarkan operasi matematis, yaitu :

Identitas : Dua buah himpunan fuzzy A dan B dikatakan sama jika keduanya didefinisikan pada suatu semesta (*universe*) dan membership function yang sama, dinyatakan dengan notasi matematis : $\mu_A(u) = \mu_B(u)$; $u \in U$ (U adalah notasi dari Universe).

Gabungan : Adalah gabungan dua buah himpunan fuzzy A dan B yang membentuk sebuah himpunan fuzzy dimana fungsi keanggotaan himpunan tersebut dinyatakan dengan notasi : $\mu_{A\cup B} = \max\{\mu_A(u), \mu_B(u)\}$ untuk semua $u \in U$. Operasi ini ditunjukkan oleh operand "**or**" dalam aturan fuzzy.

Irisan : Adalah irisan dua buah himpunan fuzzy A dan B yang membentuk sebuah himpunan fuzzy dimana fungsi keanggotaan

[1] Jun Yan, Michael Ryan, James Power, USING FUZZY LOGIC (Prentice Hall, 1994), p. 18.

himpunan tersebut dinyatakan dengan notasi : $\mu_{A\cap B} = \min\{\mu_A(u), \mu_B(u)\}$, untuk semua $u \in U$. Operasi ini ditunjukkan oleh operator "**and**" dalam aturan fuzzy.

Komplemen : Adalah himpunan fuzzy yang mempunyai fungsi keanggotaan yang dinyatakan dalam notasi : $\mu_{A'}(u) = 1 - \mu_A(u)$, untuk semua $u \in U$. Notasi $\mu_A(u)$ adalah untuk derajat keanggotaan himpunan fuzzy A dan $\mu_{A'}(u)$ adalah derajat keanggotaan himpunan fuzzy komplomen A. Operasi ini dinyatakan oleh operator "**not**" dalam aturan fuzzy.

2.2.3. SISTEM KONTROL FUZZY

Pembagian tugas pada sistem kontrol fuzzy terdiri tiga (3) tahap proses, yaitu :

(1) Proses **fuzzification**, yaitu mengubah variabel input menjadi bentuk variabel yang berarti dalam sistem fuzzy. Variabel input, berupa variabel *crisp* yang berorientasi numerik (*crisp input*), diubah ke bentuk variabel fuzzy (*fuzzy input*) dalam set fuzzy.

(2) Proses **evaluasi aturan (*rule evaluation*)**, yaitu mencari nilai aksi (*action*) dengan memberi bobot pada setiap aturan yang diberikan. Di dalam proses ini, terdapat dua komponen utama, yaitu himpunan aturan (*rule sets*) dan metode evaluasi aturan. Pada himpunan aturan, diperlukan semua aturan untuk menentukan tanggapan terhadap input atau kombinasi input yang diberikan. Sedangkan metode evaluasi aturan adalah metode yang digunakan dalam

mengevaluasi aturan yang ditetapkan. Beberapa metode evaluasi aturan yang sering dipakai adalah mini rule (Mamdani), product rule (Larsen), Max-Min rule (Zadeh), Arithmatic rule (Zadeh), dan Boolean[2].

(3) Proses **defuzzification**, merupakan tahap terakhir proses logika fuzzy yang berfungsi mengubah output yang berupa output fuzzy menjadi output crisp. Ada beberapa metode yang digunakan, yaitu Center Of Gravity (COG), fuzzy singleton, accumulate, dan intermediate. Metode accumulate berarti nilai output sama dengan nilai aksi aturan yang menang ditambah dengan nilai output sebelumnya, sehingga metode ini dapat dipakai untuk pendekatan proses integrasi. Sedangkan metode immediate pada output berarti nilai output sama dengan nilai aksi aturan yang menang.

2.2.4. IC *FUZZY LOGIC* KONTROLER AL220

IC Fuzzy Mikrokontroler AL220 adalah sebuah mikrokontroler yang khusus didesain untuk memproses sinyal input dan menghasilkan output berdasarkan logika Fuzzy. IC AL220 ini bekerja secara *stand alone* dalam melakukan perhitungan aksi output secara hardware sesuai dengan kondisi input. Spesifikasi AL220 adalah sebagai berikut[3] :

- Mempunyai 8 kanal input dan output analog maupun digital 8-bit yang bekerja secara time division multiplexing.

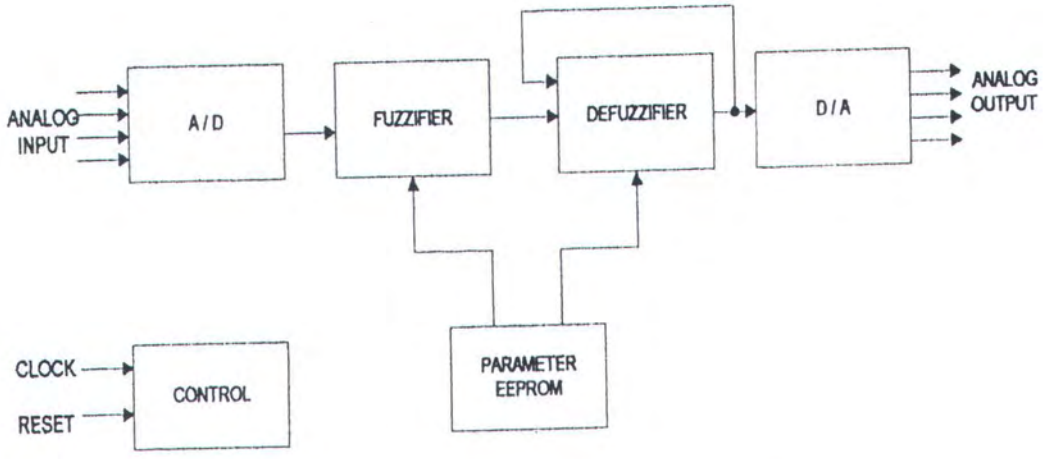
---

[2] Ibid., p. 32.
[3] Data Sheet AL220, BasiConcepts, p.1

- Mempunyai 6 buah tipe Membership Function, yaitu Left Inclusive, Symmetrical Inclusive, Right Inclusive, Symmetrical Exclusive, Left Exclusive, dan Right Exclusive.
- Mempunyai internal memori EEPROM sebagai tempat data rule.
- Mampu memproses hingga 111 Fuzzifier dengan jumlah rule maksimal sebanyak 50 buah rule.

A. BLOK DIAGRAM AL220

Komponen utama AL220 adalah **Fuzzifier**, **Deffuzzifier**, dan **Controller**. Data input digital disimpan dalam input latch, kemudian fuzzifier membandingkan data input dengan MF untuk menghitung nilai dari setiap variabel fuzzy yang berkorelasi dengan input tertentu (*winning rule*) berdasarkan metode MAX-of-MIN (yang paling mendekati dari yang tidak mendekati). Sedangkan input analog harus dikonversi terlebih dulu ke digital sebelum di-latch. Bagian Deffuzzifier akan menentukan nilai aksi winning rule dan me-latch untuk dikonversi ke output analog atau internal feedback.



Gambar 2.2. Blok Diagram IC Fuzzy AL220[4]

B. FUNGSI PIN

Fungsi pin-pin AL220 dapat dikelompokkan sebagai berikut:

Pin-pin Input

RESET : sinyal untuk menginisialisasi AL220. Pin reset harus tetap aktif low selama minimal 8 clock cycles.

AIN(3:0) : Empat buah kanal input data analog yang akan dikonversi ke 8 bit data digital oleh ADC internal. Input yang tidak digunakan harus dihubungkan ke ground.

XIN : input clock, dapat menggunakan clock eksternal atau menggunakan kristal yang salah satu ujungnya dihubungkan ke ground.

---

[4] Ibid., p.4

PRESCALE : kondisi high ('1') pada pin ini menandakan dalam mode prescale dan low '0' dalam operasi normal. Pin ini di-ground-kan saat mode prescale tidak digunakan atau dihubungkan dengan pin READY untuk operasi kontinyu. Mode juga bisa dipanggil selama pengoperasian oleh logika ekternal. Setelah RESET diaktifkan, PRESCALE input harus dipertahankan pada logika low ('0') minimal selama 4 clock cycles.

Pin-pin Output

AOUT(3:0) : Empat buah kanal output untuk tegangan analog hasil konvesi DAC internal.

READY : setelah reset, pin ini mengirim sinyal yang menunjukkan telah siap menyampling dan akan memproses data. Selama operasi, pin ini dibiarkan tidak terhubung atau disambungkan ke PRESCALE.

VREF : memfilter tegangan referensi internal. Pin ini dihubungkan ke ground melalui sebuah kapasitor 0,1 mikro Farad.

C. VARIABEL FUZZY

Adalah ekspresi linguistic hubungan input dengan membership function yang diperoleh dari proses fuzzifier dan menghasilkan input fuzzy dengan derajat keanggotaan tertentu, dalam hal ini nilainya 0-63.

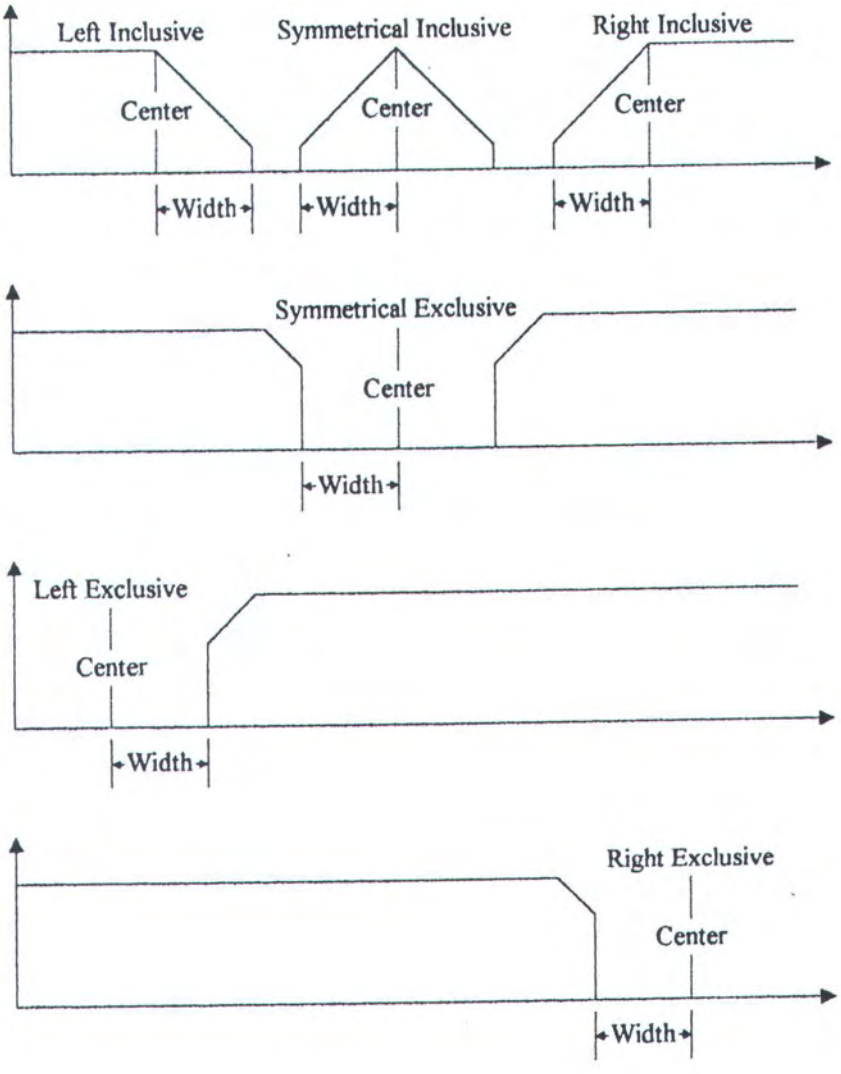
Contoh variabel fuzzy ini adalah: *Error is Small*

D. MEMBERSHIP FUNCTION

Membership function atau fungsi keanggotaan menyatakan seberapa besar derajat keanggotaan suatu input dengan variabel fuzzy tertentu atau seberapa dekat suatu input tersebut dengan nilai tengah (centre) dari suatu variabel fuzzy.

Untuk menyatakan atau merumuskan suatu fungsi keanggotaan diperlukan nilai pusat (centre) dan lebar (width) dari suatu variabel fuzzy. Bentuk dari membership function itu tergantung dari fuzzy logic controller yang digunakan.

AL220 menyediakan 6 buah tipe membership function, yaitu: Left Inclusive; Symmetrical Inclusive; Right Inclusive; Symmetrical Exclusive; Left Exclusive; dan Right Exclusive, seperti terlihat pada gambar berikut :



Gambar 2.3. Tipe-tipe Membership function AL220

Nilai center dan width MF (memberhip function) dapat dibuat konstan maupun variabel dengan fasilitas floating, dimana nilai center dan width-nya dapat diambil dari salah satu input atau output.

Misalnya : *IN1 is small (0,25, Symmetrical inclusive)*

IN2 is small (0,25, Symmetrical inclusive)

Output +1 if IN1 is small and IN2 is small

Karena IN1 dan IN2 memiliki MF yang sama maka rule konvensional ini dapat digantikan sebagai berikut :

IN1 is small_diff (IN2, 0, Symmetrical Exclusive)

Output +2 if IN1 is small_diff

IN2 disimpan di input latch sebagai nilai center IN1. Keuntungan dari rule dengan floating di atas adalah dapat menghemat memori, karena menggunakan lebih sedikit variabel fuzzy dan rule.

E. RULES

Sebuah rule terdiri dari satu atau lebih variabel fuzzy dan sebuah action value keluaran. Rule digunakan untuk memberitahu pengontrol bagaimana merespon data input yang ada.

Pada contoh di bawah ini setiap rule mengandung dua variabel fuzzy. Rule dimasukkan ke INSiGHT dalam format berikut:

Output = -5 if Velocity is Fast and Acceleration is Positive

Output = +5 if Velocity is Little_Slow and Acceleration is Zero

Pada rule pertama, variabel fuzzy pertama ialah **Velocity is Fast** variabel fuzzy kedua ialah **Acceleration is Positive.** Aksi –5 ialah nilai numerik yang diterapkan di output untuk mempercepat atau memperlambat motor.

F. FLOATING MEMBERSHIP FUNCTION

Salah satu hal istimewa yang dimiliki oleh AL220 ialah mempunyai floating membership function. Floating membership function mempunyai nilai center dan width yang berubah-ubah.

Floating membership function mengubah nilai center dan width sebagai data dari input yang dipilih atau perubahan output.

Sebagai contoh, dua variabel fuzzy dengan membership function yang terdapat dalam tanda kurung dapat didefinisikan secara konvensional dan digunakan dalam rule berikut:

IN1 is small (0,25,Symmetrical Inclusive)

IN2 is small (0,25,Symmetrical Inclusive)

di mana angka pertama yaitu 0 menunjukkan center dan angka kedua yaitu 25 menyatakan width dari membership function.

Output = +1 if IN1 is small and IN2 is small

Di mana variabel fuzzy **IN1 is small** membandingkan input IN1 dengan membership function konvensional **small**.

Floating membership function membuat deskripsi yang sama tetapi lebih singkat seperti yang terdapat dalam variabel fuzzy dan rule di bawah ini:

IN is small_difference (IN2,25,Symmetrical Exclusive)

Output = +1 if IN1 is small_difference

Pada variabel fuzzy, center dari membership function small_difference didefinisikan oleh nilai IN2 yang tersimpan pada latch input. Dalam proses fuzzifikasi, sebuah input dikurangkan dari center dari membership function dan

hasilnya dibalik untuk menentukan seberapa dekat input tersebut dengan nilai center. Bila center dari suatu membership function berupa floating center, maka suatu input akan dikurangkan ke input yang lain.

Floating membership fuunction memungkinkan untuk menggunakan suatu variabel fuzzy yang secara langsung mengukur perbedaan dari dua input. Teknik ini dapat digunakan sebagai contohnya untuk mengkalibrasi perubahan pada sensor setiap waktu.

G. *FUZZIFIER*

Fuzzifier membandingkan data input latch dengan *membership function* untuk menghitung nilai variabel fuzzy. Ketika penghitungan MIN rule dilakukan, nilainya mewakili rule yang disimpan. Ketika penghitungan MAX dilakukan pada seluruh variabel yang mereferensikan nilai output, nilai rule pemenang akan diberikan ke Defuzzifier.

Rule akan dievaluasi menurut urutan rule tersebut dimasukkan. Setiap rule dapat mengacu ke segala output. Output dapat direferensikan berulang dalam suatu set rule.

H. PENG-UP DATE-AN OUTPUT LATCH

Rule dievaluasi dalam urutan saat masuknya. Banyak rule dapat mereferensikan output berulang-ulang di dalam sebuah set rule. Ketika sebuah rule atau grup rule memberikan output yang dievaluasi dan rule selanjutnya memasukkan referensi ke output lain, kompiler akan menyertakan kode untuk rule

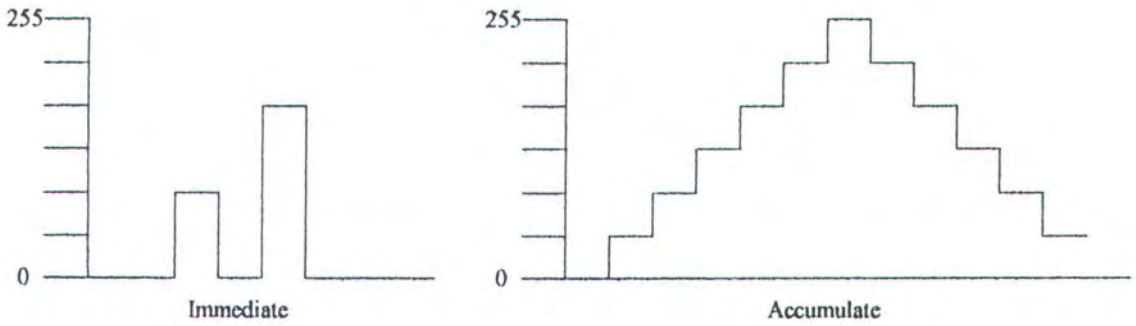
terakhir dengan output *latch* untuk di-*update* dengan nilai pemenang yang baru. *Latch* data juga bisa dengan cepat dipakai sebagai *feedback*.

Jika setelah pemrosesan rule yang berefek ke output lain, prosessor menemukan rule atau grup rule lain yang menunjuk output sebelumnya, output latch akan di-update lagi. Peng-update-an output mungkin bisa sesering mungkin selama proses sebagaimana disana ada bagian grup terpisah yang mereferensikannya.

Sebagaimana sebelumnya, sampling input adalah kontinyu. Output Analog juga sering di-update terus menerus. Selama proses variabel fuzzy mungkin memakai data sample yang lalu atau dari data yang sedang dipakai proses, tergantung dimana sampling input cycle berada relatif terhadap prosessing cycle. Jika lebih dari satu grup rule yang mereferensi ke input dan output yang sama, maka nilai output akan berubah lebih dari satu kali selama sebuah proses cycle berdasar pada perbedaan input data.

I. METODE DEFUZZIFIKASI

AL220 memiliki dua metode defuzzifikasi, yaitu **Accumulate** dan **Immediate**. Metode immediate, nilai output sama dengan nilai aksi *winning rule*. Sedangkan metode accumulate, nilai outputnya merupakan nilai aksi winning rule ditambah nilai output sebelumnya.



Gambar 2.4. Metode Defuzzifikasi

J. ORGANISASI MEMORI

AL220 memiliki 256 byte lokasi memori untuk menyimpan parameter aplikasi. 32 byte terakhir untuk menyimpan nilai center dan width yang tetap. 224 byte lainnya untuk menyimpan rule. Setiap rule membutuhkan dua byte ditambah dua byte lagi setiap penggunaan satu variabel fuzzy. Organisasi memori ini seperti ditunjukkan pada tabel berikut:

Tabel 2.1 Organisasi Memori AL 220[5]

Alamat (Desimal)	Alamat (Heksamal)	Fungsi
0	00	Rules
⟷	⟷	⟷
223	DF	Rules
224	E0	Centers
⟷	⟷	⟷
239	EF	Centers
240	F0	Width
⟷	⟷	⟷
255	FF	Width

Byte pertama pembentuk variabel fuzzy disimpan pada *even address*, byte kedua pada *odd address*. Byte-byte ini untuk mengontrol kerja AL220.

[5] Ibid., p.8.

Tabel 2.2. Command Byte dan Select Byte AL220[6]

Command Byte / Alamat genap

7	6	5	4	3	2	1	0
WF	CF	I/O cont		I/O select	type	2-7	
AF	Mode				type	1	
AF	Mode			output select	type	0	

Select Byte / Alamat Ganjil

7	6	5	4	3	2	1	0
Center select				Width select			
	I/O cont	I/O select center			I/O cont		I/O select width
ACTION							
					I/O cont	I/O select action	

Type 2 1 0
- 0 0 0 Last Term of last Rule of given output
- 0 0 1 Last Term of Current Rule
- 0 1 0 MF, Symmetrical, Inclusive
- 0 1 1 MF, Symmetrical, Exclusive
- 1 0 0 MF, Left, Inclusive
- 1 0 1 MF, Left, Exclusive
- 1 1 0 MF, Right, Inclusive
- 1 1 1 MF, Right, Exclusive

I/O Select 4 3
- 00 I/O port 0 as Input
- 01 I/O port 1 as Input
- 10 I/O port 2 as Input
- 11 I/O port 3 as Input

I/O Control 5
- 0 Select from Inputs
- 1 Select from outputs

Mode 6
- 0 Immediate, Output equals Action
- 1 Accumulate, Output equals current output plus two's complement Action (-128 to +127)

AF 7
- 0 Select Action from select Byte (FIXED)
- 1 Select Action from I/O via select Byte (FLOAT)

Ouput Select 4 3
- 00 ACTION from current RULE set to Output 0
- 01 ACTION from current RULE set to Output 1
- 10 ACTION from current RULE set to Output 2
- 11 ACTION from current RULE set to Output 3

CF 6
- 0 Select Center from memory via Select byte (FIXED)
- 1 Select Center from I/O via Select byte (FLOAT)

WF 7
- 0 Select Width from memory via Select byte (FIXED)
- 1 Select Width from I/O via Select byte (FLOAT)

Width Select (0 - 3)
Used as address index (E0-EF) for fixed 6-bit value Width when type= 2-7 and WF= 0.

Center Select (4 - 7)
Used as Address index (F0-FF) for fixed 8-bit CENTER value when type = 2-7 and CF = 0

I/O Select Width 1 0
- 00 I/O port 0 as Width (Type=2-7 and WF=1)
- 01 I/O port 1 as Width (Type=2-7 and WF=1)
- 10 I/O port 2 as Width (Type=2-7 and WF=1)
- 11 I/O port 3 as Width (Type=2-7 and WF=1)

I/O Control 2
- 0 Select from Inputs (Type = 2-7 and WF=1)
- 1 Select from outputs (Type = 2-7 and WF=1)

I/O Select Center 5 4
- 00 I/O port 0 as Input (type=2-7 and WF=1)
- 01 I/O port 1 as Input (Type=2-7 and WF=1)
- 10 I/O port 2 as Input (Type=2-7 and WF=1)
- 11 I/O port 3 as Input (Type=2-7 and WF=1)

I/O Control 6
- 0 Select from Inputs (Type = 2-7 and WF=1)
- 1 Select from outputs (Type = 2-7 and WF=1)

ACTION (0 - 7)
8-Bit Action value to be applied to an output due to winning of Last Term of Last Rule (Type = 1)
or
Last Term of Last Rule of given output (Type = 0), and AF = 0 (Fixed)

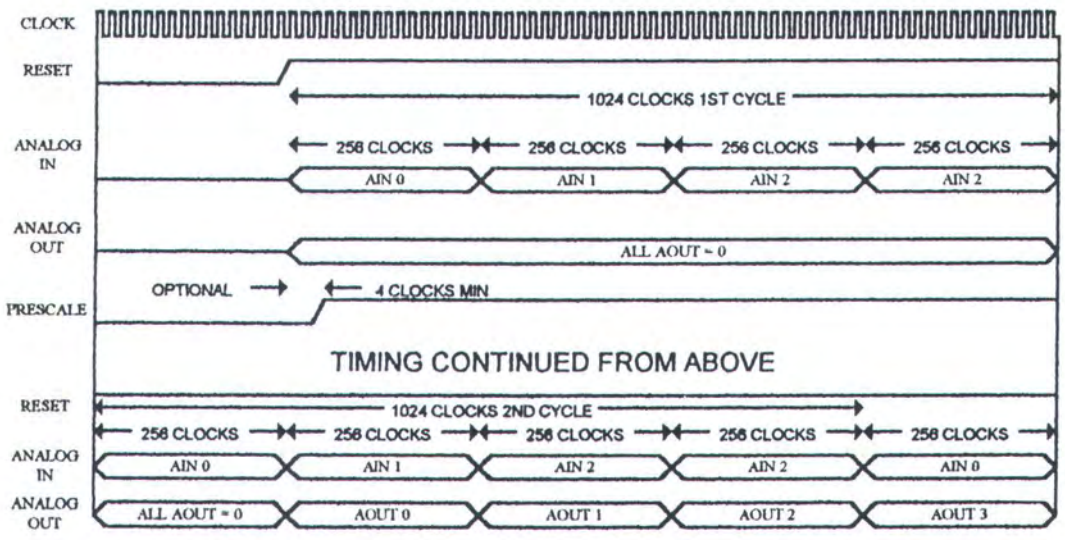
I/O Select Action 1 0
- 00 I/O port 0 as Action (type=0-1 and AF=1)
- 01 I/O port 1 as Action (type=0-1 and AF=1)
- 10 I/O port 2 as Action (type=0-1 and AF=1)
- 11 I/O port 3 as Action (type=0-1 and AF=1)

I/O Control 2
- 0 Select from Inputs (Type = 0 - 1and WF=1)
- 1 Select from outputs (Type = 0 - 1 and WF=1)

[6] Ibid., p.9

K. SISTEM TIMING AL220

Sistem timing pada AL220 dapat dilihat pada gambar berikut,



Gambar 2.5. Timing Diagram AL220[7]

Pada saat mulai bekerja, pin RESET diaktifkan sehingga semua latches direset menjadi low, output digital berlogic low, dan analog output tetap pada level sebelumnya saat reset dilakukan. Jika RESET diaktifkan selama seratus cloks atau lebih, input analog akan menjadi nol saat sampling dilakukan. Jika RESET diaktifkan kurang dari seratus clock, mungkin ada sisa data dari sampel terakhir pada analog input. Saat RESET tidak diaktifkan AL220 mulai melakukan proses sampling input selama 1024 clock yang pertama.

Nilai input analog dikonversi menjadi data digital dan dilatch secara internal dalam periode 256 clocks masing-masing secara berurutan. Total 1024

[7] Ibid., p.10

clock diperlukan untuk mengkonversi keempat input sesudah proses konversi tersebut berulang. Pada clock maksimum waktu sampling untuk setiap input ialah 10 Khz atau 100 ms.

Waktu 1024 clock yang pertama untuk proses dimulai setelah proses konversi untuk seluruh input telah selesai. Waktu proses terdiri dari 1024 clock tidak tergantung dari jumlah variabel fuzzy dan jumlah rule yang digunakan.

Evaluasi variabel fuzzy dan rule yang ada memerlukan masing-masing empat clock. Sebagai contohnya, sebuah rule dengan dua fuzzy variabel akan memerlukan 12 clock untuk proses. Selama satu siklus proses setiap sebuah fuzzy variabel atau rule masing-masing diproses empat periode clock kecuali 64 periode laten clock pada akhir proses siklus.

Bila data pada latch output diumpan balikkan secara di dalam, data tersebut akan tertinggal dibelakang analog input selama 1024 clock dari inisial siklus sampling. Setelah itu, seiring dengan latch output diperbarui selama pemrosesan data umpan balik digunakan sebagai input.

Pada mode prescale, prosesor tidak aktif untuk periode 1024 clock sesudah counter bertambah. Bila counter mencapai nilai FF prosesor diaktifkan untuk satu periode 1024 clock untuk melaksanakan perhitungan fuzzy dan counter berjalan lagi. Pin prescale dapat dihubungkan ke pin Ready saat operasi prescale diinginkan.

2.3. FILTER ANALOG

Filter analog ialah suatu rangkaian analog yang melewatkan atau meredam sinyal masukan dengan frekuensi tertentu. Rangkaian ini dapat menggunakan komponen pasif seperti resistor, kapasitor, dan induktor atau rangkaian aktif yaitu dengan penguat solid state seperti operasional amplifier ditambah dengan sedikit komponen pasif seperti resistor dan kapasitor.

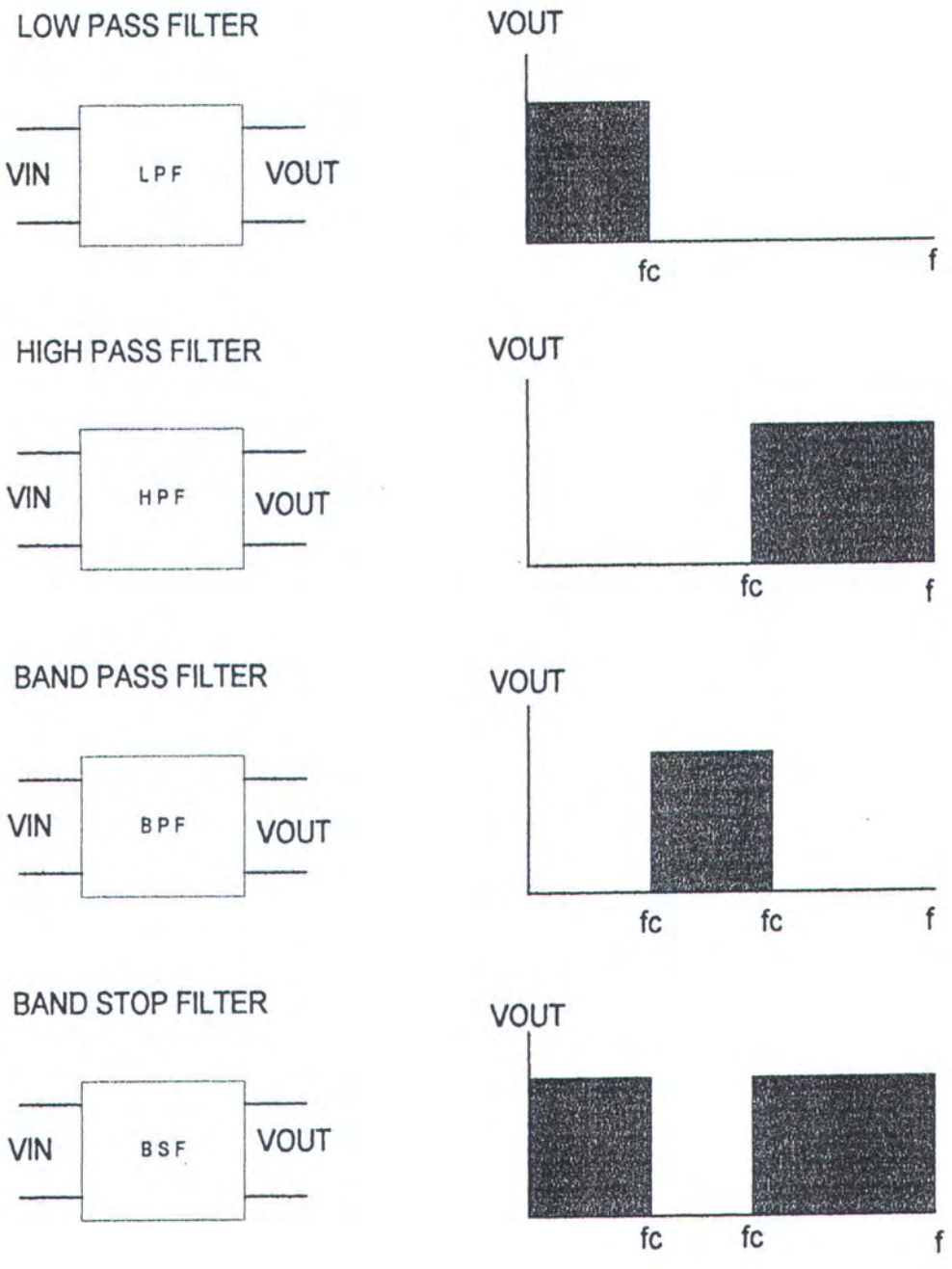
Kelemahan rangkaian pasif ialah selain ukurannya cenderung lebih besar, energi yang dibuang cukup banyak. Sebaliknya rangkaian aktif lebih kompak dan bisa memperkuat atau paling tidak mempertahankan amplitudo sinyal masukan.

2.3.1. Jenis Filter Analog

Menurut frekuensi yang dilewatkannya ada empat jenis filter analog, yaitu:

- low pass filter (LPF)
- high pass filter (HPF)
- band pass filter (BPF)
- band stop filter (BSF)

Low pass filter hanya meloloskan sinyal input yang frekuensinya lebih kecil dari frekuensi cutoff. Sebaliknya high pass filter meloloskan sinyal yang frekuensinya lebih besar dari frekuensi cutoff. Band pass filter meloloskan sinyal yang frekuensinya di antara kedua frekuensi cutoff-nya. Sedangkan band stop filter menahan sinyal di antara kedua frekuensi cutoff-nya dan melewatkan sinyal yang frekuensinya di luar batas frekuensi cutoff-nya. Respon frekuensi masing-masing filter seperti pada gambar berikut ini:



Gambar 2.6. Respon Frekuensi Filter Analog

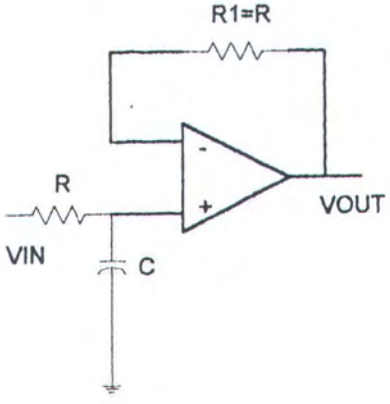
2.3.2. Rangkaian dan Fungsi Transfer Filter Analog

Fungsi transfer merupakan ekspresi yang menyatakan hubungan antara output suatu sistem terhadap input yang diberikan. Dalam menyatakan suatu

fungsi transfer digunakan persamaan matematis yang berupa persamaan bilangan kompleks.

A. Butterworth Low Pass Filter

Rangkaian Butterworth Low Pass Filter –20dB/dekade ialah sebagai berikut:



Gambar 2.7. Rangkaian LPF 20dB/dekade

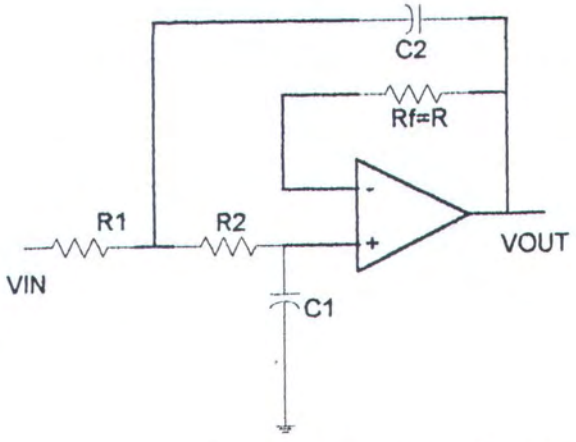
Persamaan matematis yang menghubungkan output dengan input ialah:

$$V_o = \frac{1}{1 + j\omega CR} V_i$$

$$f_c = \frac{1}{2\pi RC}$$

Di mana f_c adalah harga frekuensi cut off dari filter.

Rangkaian Butterworth Low Pass Filter –40dB/dekade ialah sebagai berikut:



Gambar 2.8. Rangkaian LPF 40dB/dekade

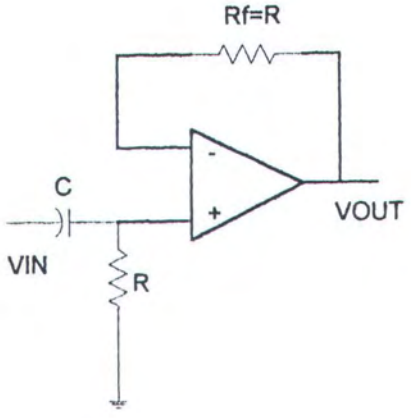
Persamaan matematis yang menghubungkan output dengan input ialah:

$$V_o = \frac{1}{1 - \omega^2 C_1^2 R^2 + j2\omega C_1 R} V_i$$

$$f_c = \frac{1}{2\pi\sqrt{2}\, C_1 R}$$

B. Butterworth High Pass Filter

Rangkaian Butterworth High Pass Filter –20dB/dekade ialah sebagai berikut:



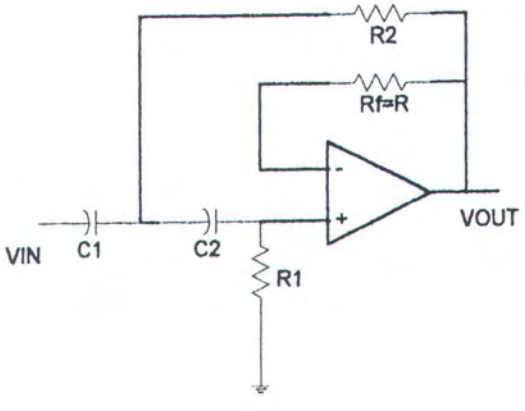
Gambar 2.9. Rangkaian HPF 20dB/dekade

Persamaan matematis yang menghubungkan output dengan input ialah:

$$V_o = \frac{1}{1 + j\frac{1}{\omega CR}} V_i$$

$$f_c = \frac{1}{2\pi RC}$$

Rangkaian Butterworth Low Pass Filter –40dB/dekade ialah sebagai berikut:



Gambar 2.10. Rangkaian HPF 40dB/dekade

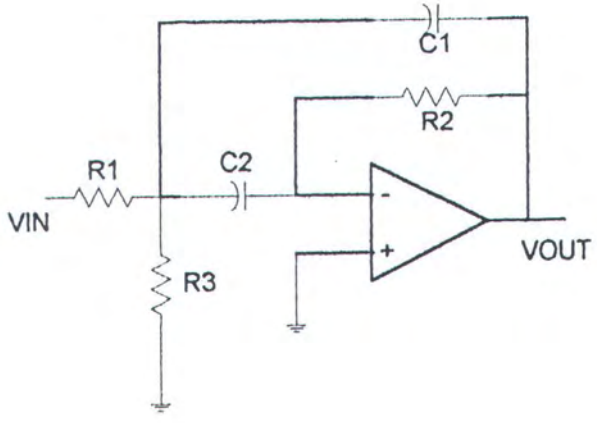
Persamaan matematis yang menghubungkan output dengan input ialah:

$$V_o = \frac{1}{1 - \frac{1}{\omega^2 C_1 C_2 R_1 R_2} + j\frac{1}{\omega C_1}\left(\frac{1}{C_1} + \frac{1}{C_2}\right)} V_i$$

$$f_c = \frac{\sqrt{2}}{2\pi R_1 C}$$

C. Band Pass Filter

Rangkaian Band Pass Filter ialah sebagai berikut:



Gambar 2.11. Rangkaian BPF

Persamaan matematis yang menghubungkan output dengan input ialah:

$$V_o = \frac{\frac{1}{R_1}}{\frac{1}{R_2} + \frac{C_1}{C_2 R_2} - j\left(\frac{1}{\omega C_2 R_1 R_2} + \frac{1}{\omega C_2 R_2 R_3} - \omega C_1\right)} V_i$$

Bila $C_1 = C_2$ maka persamaan di atas menjadi:

$$V_o = \frac{\frac{1}{R_1}}{\frac{2}{R_1} - j\left(\frac{1}{\omega C R_1 R_2} + \frac{1}{\omega C R_1 R_3} - \omega C\right)} V_i$$

$$f_c = \frac{1}{2\pi C}\sqrt{\frac{R_1 + R_3}{R_1 R_2 R_3}}$$

Pada tugas akhir ini filter yang digunakan ialah band pass filter. Tahap perencanaan filter ini ialah sebagai berikut:

1. Dipilih faktor penguatan (G) yang diinginkan, dan dua di antara frekuensi resonansi (f_r), bandwidth (BW), dan faktor kualitas (Q). Variabel yang belum ditentukan dicari dengan persamaan $2\pi f_\rho$ = BWxC.
2. Dipilih $C_1 = C_2 = C$ (antara 0,001uF sampai 0,1 uF)
3. Harga R_2 dicari dari persamaan $R_2 = 2 / (BWxC)$
4. Harga R_1 dicari dari persamaan $R_1 = R_2 / (2xG)$
5. Harga R_3 dicari dari persamaan $R_3 = R_2 / (4Q^2-2G)$

2.4. TRANSKONDUKTANSI AMPLIFIER

Salah satu transkonduktansi amplifier produksi National Semiconductor ialah LM13600. Seri LM13600 ini terdiri dari dua transkonduktasi amplifier yang dikontrol oleh arus dengan diferensial input dan pushpull output. Kedua amplifier itu memakai sumber tegangan yang sama tetapi beroperasi secara terpisah. Dioda pelinier disediakan pada bagian input untuk mengurangi distorsi dan memungkinkan level input yang lebih tinggi. Terdapat juga buffer yang impedansinya terkontrol yang khusus didesain untuk melengkapi range dinamik dari amplifier.

Spesifikasi LM13600 adalah sebagai berikut: [8]

- G_m dapat diatur diatas 6 dekade
- Mempunyai G_m yang linier
- Mathcing yang sempurna antar amplifier
- Diode pelinier
- Impedansi buffer yang terkontrol
- Signal to ratio output yang tinggi.

Aplikasi dari LM13600 ialah:

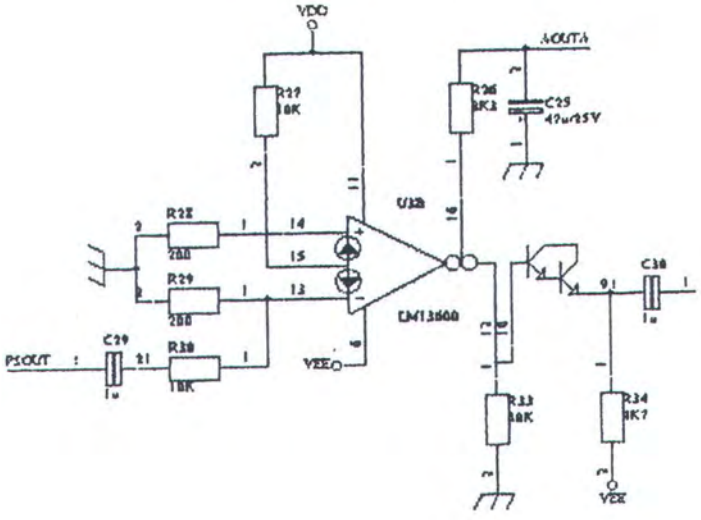
- Penguat terkontrol oleh arus
- Impedansi terkontrol oleh arus
- Filter terkontrol oleh arus
- Oscillator terkontrol oleh arus
- Multiplekser
- Pewaktu
- Rangkaian sample and hold

2.4.1. Rangkaian Penguat Terkontrol Tegangan

Besarnya faktor penguatan pada rangkaian ini tergantung dari besarnya tegangan pengontrol yang diberikan. Semakin besar tegangan pengontrol maka semakin besar pula penguatan yang diberikan terhadap sinyal masukan. Sinyal keluaran yang dihasilkan ialah hasil penguatan dari sinyal input yang masuk,

[8] Data Sheet LM13600, National Semiconductor, p.1

dalam hal ini yang berubah ialah amplitudo sinyal masukan, sedangkan frekuensinya tetap. Berikut ini rangkaian penguat terkontrol tegangan:

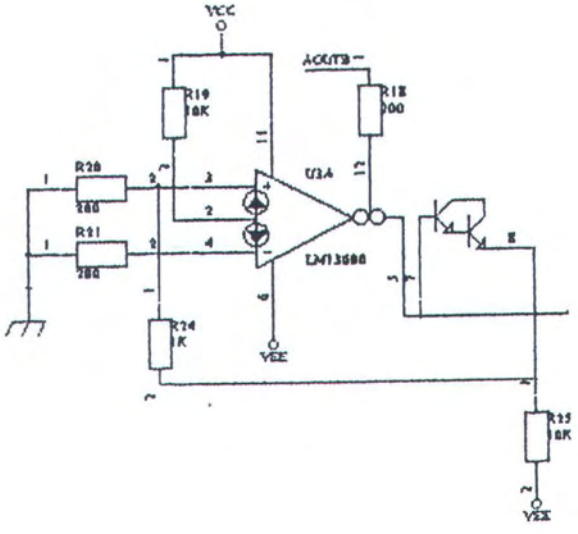


Gambar 2.11. Rangkaian penguat terkontrol tegangan

AOUTA merupakan sinyal pengontrol dan PSOUT merupakan sinyal input yang akan diperkuat.

2.4.2. Rangkaian Tahanan Terkontrol Tegangan

Sebuah operasional transkonduktansi amplifier dapat digunakan untuk menghasilkan resistor atau tahanan yang besarnya tergantung dari tegangan kontrol yang diberikan. Rangkaian ini berguna untuk mengkonversi besaran tegangan menjadi besaran tahanan. Rangkaian tahanan terkontrol tegangan dapat dilihat pada gambar berikut ini:



Gambar 2.12. Rangkaian hambatan terkontrol tegangan

AOUTB merupakan sinyal pengontrol. Output rangkaian ini dihubungkan ke suatu rangkaian lain yang membutuhkan perubahan nilai hambatan yang dapat dikendalikan.

BAB III

PERANCANGAN PERANGKAT KERAS DAN PERANGKAT LUNAK

Pada bab ini akan dibahas mengenai perancangan sistem yang akan dibuat, baik itu perancangan perangkat keras maupun perangkat lunak yang digunakan. Pembahasan diawali dengan pendahuluan yang berisi tentang konsep dasar yang dipakai dalam perancangan sistem, contoh penerapan dan peralatan atau komponen yang digunakan dalam sistem.

Pembahasan tentang perancangan perangkat keras terdiri dari blok diagram sistem dengan penjelasannya, rangkaian *band pass filter* yang digunakan, rangkaian penggeser fase yang menggunakan IC LM13600, rangkaian penentu faktor penguat yang menggunakan IC LM13600, rangkaian *fuzzy logic controller* yaitu AL220, dan rangkaian penguat dan *band pass filter* untuk sinyal error.

Pembahasan tentang perancangan perangkat lunak menjelaskan perangkat lunak yang digunakan oleh *fuzzy logic controller* AL220, terdiri dari penentuan input dan output yang diperlukan, pembuatan variabel fuzzy, dan pembuatan rule yang diperlukan untuk menentukan output dari *fuzzy logic controller* AL220 terhadap sinyal input yang ada.

3.1. PENDAHULUAN

Metode untuk menghasilkan sinyal anti noise berdasarkan prinsip bahwa sebuah sinyal bila ditambahkan kepada inversnya sendiri akan menghasilkan nol karena kedua sinyal tersebut saling meniadakan. Masalahnya ialah dalam daerah waktu, menghasilkan sinyal inverse yang benar-benar tepat merupakan hal yang sangat sukar untuk dilakukan. Pengaruh pemantulan akustik dapat mempengaruhi sinyal asli dan akan menambah kekompleksan sistem.

Spesifikasi eksak untuk sinyal anti noise mungkin diketahui untuk bebarapa sistem yang periodik. Dalam hal ini hanya fase dan amplitudo yang perlu diatur. Sebuah sistem transformator atau motor yang bekerja pada frekuensi tertentu akan menghasilkan noise dengan frekuensi yang sama dengan frekuensi kerjanya ditambah dengan frekuensi harmonik yang timbul. Untuk sistem ini, frekuensi eksaknya diketahui sehingga sinyal dengan frekuensi tersebut dapat dibangkitkan lagi, digeser fasenya, dan amplitudonya diatur untuk dapat menghasilkan sinyal anti noise. Untuk mengatur faktor penguatan dari penguat sinyal anti noise dan besarnya pergeseran fase dari rangkaian penggeser fase digunakan Fuzzy Logic Controller AL220.

Untuk penyederhanaan permasalahan, sinyal error yang digunakan ialah nilai RMS dari sinyal sensor noise. Untuk menentukan nilai RMS ini digunakan juga Fuzzy Logic Controller AL220. Tanggapan waktu adalah hal yang penting, jika waktu ditentukan terlalu lambat maka transien tidak akan terlihat, jika terlalu cepat harga nol mungkin tidak akan pernah tercapai. Pengontrol AL220 akan memonitor sinyal error dan mmbangkitkan dan membangkitkan derivatifnya

sehingga arah error dapat diukur. Derivatir dapat dihasilkan dengan cara menduplikat sinyal error ke sebuah output dan pada siklus berikutnya membandingkan nilai lama yang tersimpan pada output dengan sinyal error yang baru.

Dalam bab ini khususnya akan dibahas perancangan sistem anti noise control yang menggunakan fuzzy logic. Pembahasan dibagi menjadi dua bagian, yakni perancangan perangkat keras dan perancangan perangkat lunak.

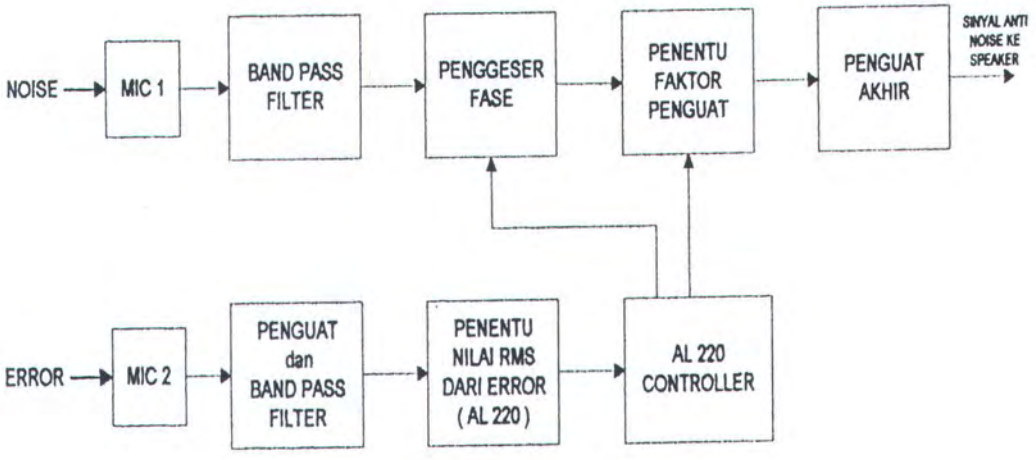
Untuk perancangan perangkat keras akan dibahas mengenai perancangan sistem dan hardware yang meliputi beberapa bagian, yakni unit sensor noise, unit sensor error, unit fuzzy logic controller AL220, dan unit penghasil sinyal anti noise.

Adapun pembahasan perangkat lunak ialah perangkat lunak untuk fuzzy logic controller AL220. Perangkat lunak ini meliputi variabel fuzzy yang ada dan rule yang digunakan untuk menghasilkan sinyal anti noise untuk melawan noise yang ada. Perangkat lunak ini diisikan ke dalam media penyimpanan EEPROM yang terdapat dalam Fuzzy Logic Controller AL220.

3.2. PERANCANGAN PERANGKAT KERAS

Perangkat keras sistem pengontrol noise terdiri dari beberapa bagian yaitu mikrofon, band pass filter, rangkaian penggeser fase, rangkaian penyesuai faktor penguatan, rangkaian penguat sinyal anti noise, dan Fuzzy Logic Controller AL220.

Blok diagram perangkat keras yang direncanakan adalah sebagai berikut:



Gambar 3.1. Blok diagram perangkat keras

Dari blok diagram di atas tampak bahwa terdapat dua sinyal input yaitu sinyal noise yang berasal dari gerakan angin karena kipas angin yang berputar dan sinyal error yang merupakan perpaduan atau superposisi antara sinyal noise dengan sinyal anti noise yang dihasilkan oleh sistem. Mikrofon 1 untuk menangkap sinyal noise diletakkan pada rumah kipas angin dan diusahakan agar mikrofon tersebut tidak menangkap sinyal anti noise. Mikrofon 2 yang digunakan untuk menangkap sinyal error diletakkan di antara kipas angin penghasil noise dan speaker penghasil sinyal anti noise.

Sinyal dari sensor noise yang masuk melalui MIC 1 dilewatkan bandpass filter. Karena kipas angin membangkitkan frekuensi primer 120 Hz, frekuensi ini

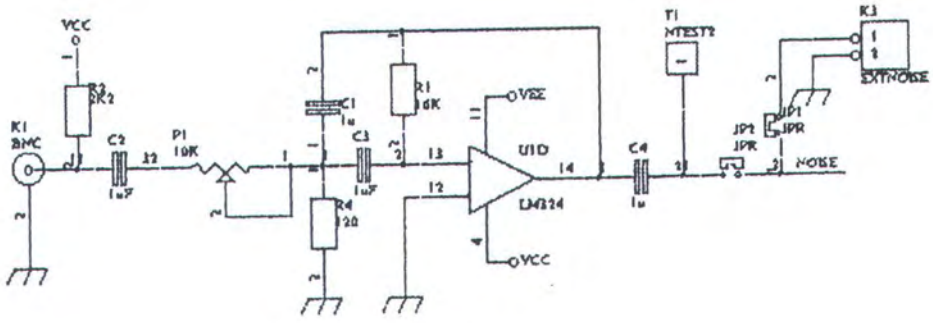
dipilih menjadi frekuensi pusat dari bandpass filter. Sinyal noise yang sudah difilter ini kemudian fasenya digeser. Besarnya pergeseran fase atau delay ini ditentukan oleh kapasitor dan resistor variabel yang dibentuk oleh transconductance amplifier. Setelah fase disesuaikan sinyal noise disesuaikan amplitudonya oleh transconductance amplifier yang kedua. Akhirnya sinyal yang sudah mengalami penyesuaian fase dan amplitudo ini dikuatkan oleh suatu penguat. Output penguat ini dihubungkan ke speaker untuk menghasilkan sinyal anti noise.

Besarnya penyesuaian fase atau penyesuaian amplitudo yang dilakukan pada sinyal noise ditentukan oleh analisa sinyal error. Sinyal error masuk sistem melalui MIC 2 kemudian diperkuat oleh suatu penguat sebelum memasuki rangkaian Bandpass Filter. Penguatan sinyal perlu dilakukan karena sinyal error semakin lama semakin kecil saat sistem bekerja. Nilai ideal sinyal error ini ialah nol. Sinyal lalu diinputkan ke Fuzzy Logic Controller AL220 untuk mengubah sinyal error menjadi RMS error. Nilai RMS sinyal error inilah yang menjadi input bagi Fuzzy Logic Controller AL220 lainnya yang bertugas mengontrol penyesuaian fase dan amplitodo dari sinyal noise.

Transconductance amplifier menerima tegangan antara –5V sampai 0V. Karena kedua transconductance amplifier ini dikendalikan langsung oleh AL220 maka kedua AL220 itu diberi tegangan antara –5V sampai 0V juga. Dengan demikian tidak diperlukan rangkaian perantara level tegangan.

3.2.1. Rangkaian Bandpass Filter

Gambar rangkaian bandpass filter yang digunakan sebagai berikut:



Gambar 3.2. Rangkaian bandpass filter

Bandpass filter yang digunakan ialah bandpass dengan frekuensi tengah 120 Hz karena noise yang dihasilkan oleh kipas angin frekuensinya 120 Hz. Faktor kualitas (Q) filter ialah 6. Sedangkan faktor penguatannya (G) 8 sampai 10. Harga komponen dapat dihitung dengan persamaan berikut ini:

$$Q = \frac{w_r}{BW}$$

$$BW = \frac{w_r}{Q} = \frac{2\pi * 120}{6} = 125,6$$

$$R_1 = \frac{2}{BW * C} = \frac{2}{125,6 * 1\mu} = 15923,5 \approx 16K$$

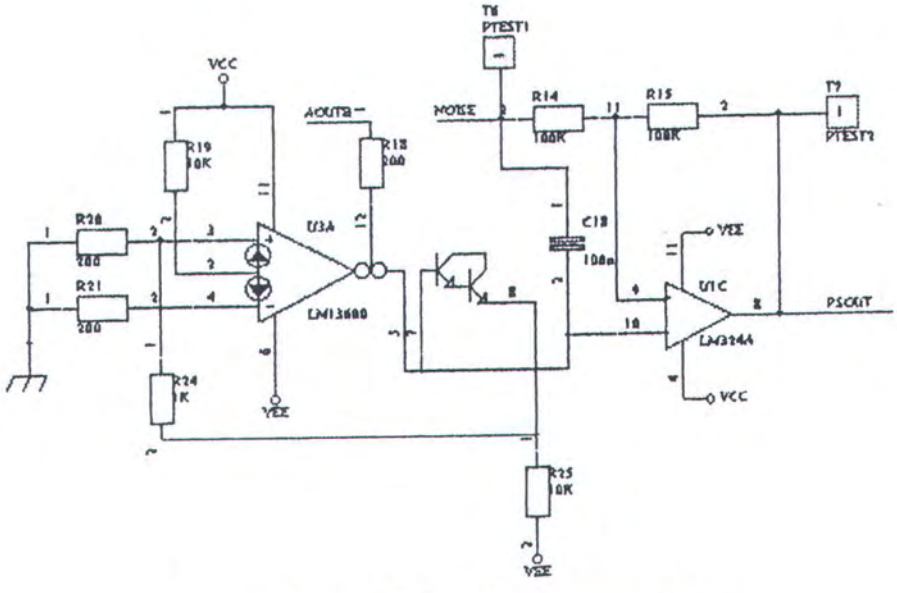
$$R_4 = \frac{R_1}{4Q^2 - 2G} = \frac{16K}{4 * 6^2 - 2 * 8} = 120$$

$$D_1 = \frac{R_1}{2G} = \frac{16K}{2 * 8} = 1K$$

Sinyal input dari bandpass filter ini berasal dari mikrofon 1 yang diletakkan pada rumah kipas angin. Sinyal ini berupa sinyal noise yang ditimbulkan oleh udara bergerak di sekitar kipas angin. Sinyal keluaran dari filter ini akan dimasukkan ke rangkaian penggeser fase.

3.2.2. Rangkaian Penggeser Fase

Bagian ini berguna untuk menggeser fase dari sinyal input. Rangkaian penggeser fase ialah sebagai berikut:



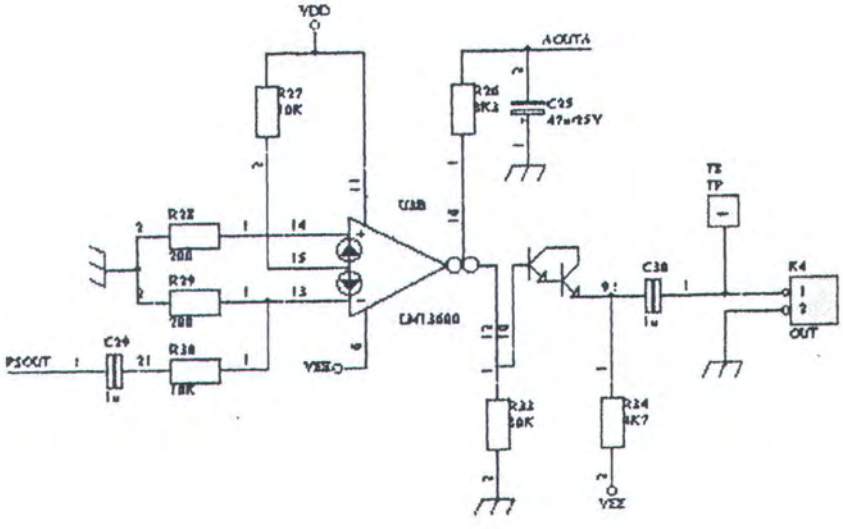
Gambar 3.3. Rangkaian penggeser fase

Besarnya pergeseran fase yang dilakukan terhadap sinyal input tergantung dari harga kapasitor C18 dan resistor variabel yang dibentuk oleh transconductance amplifier LM13600N. Sedangkan transconductance amplifier ini dikendalikan oleh sinyal phase (AOUTB) yaitu sinyal keluaran dari Fuzzy Logic Controller AL220 yang menentukan besarnya pergeseran fase.

Setelah mengalami penyesuaian fase maka sinyal akan disesuaikan amplitudonya.

3.2.3. Rangkaian Penentu Faktor Penguat

Bagian ini berguna untuk menentukan faktor penguatan yang diberikan kepada sinyal keluaran dari rangkaian penggeser fase. Rangkaian penentu faktor penguat ialah sebagai berikut:



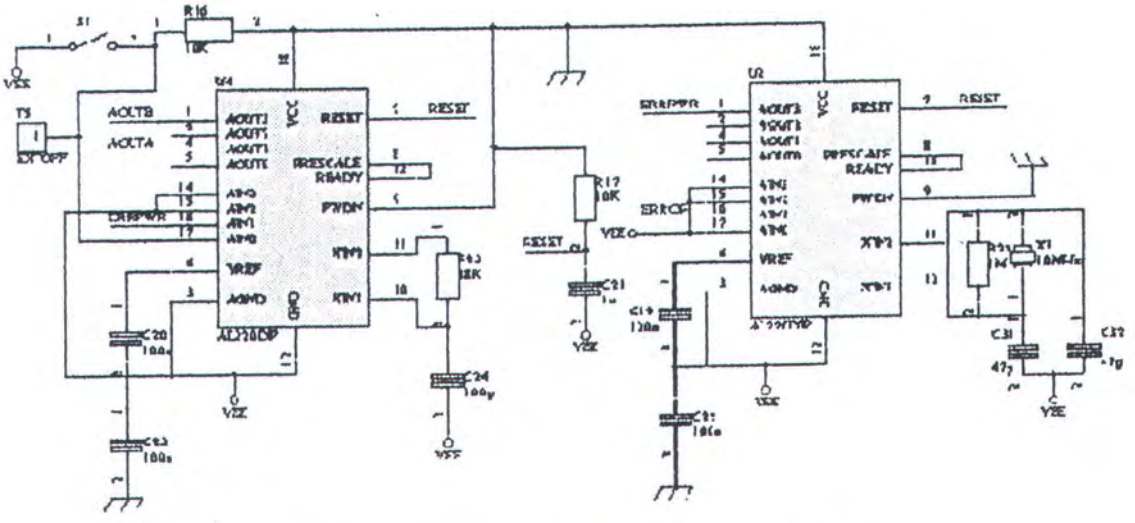
Gambar 3.4. Rangkaian penentu faktor penguat

Besarnya penguatan yang dilakukan terhadap sinyal keluaran dari rangkaian penggeser fase tergantung dari harga resistor variabel yang dibentuk oleh transconductance amplifier LM13600N. Sedangkan transconductance amplifier ini dikendalikan oleh sinyal gain (AOUTA) yaitu sinyal keluaran dari Fuzzy Logic Controller AL220 yang menentukan besarnya penguatan.

Setelah mengalami penyesuaian amplitudo maka terbentuklah sinyal anti noise. Sinyal ini yang kemudian dikuatkan oleh penguat akhir dan dikeluarkan oleh speaker.

3.2.4. Rangkaian Fuzzy Logic Controller AL220

Rangkaian fuzzy logic controller AL220 ini merupakan bagian utama untuk menentukan besarnya pergeseran fase dan faktor penguatan yang diberikan ke sinyal noise. Rangkaiannya ialah sebagai berikut:



Gambar 3.5. Rangkaian Fuzzy Logic Controller AL220

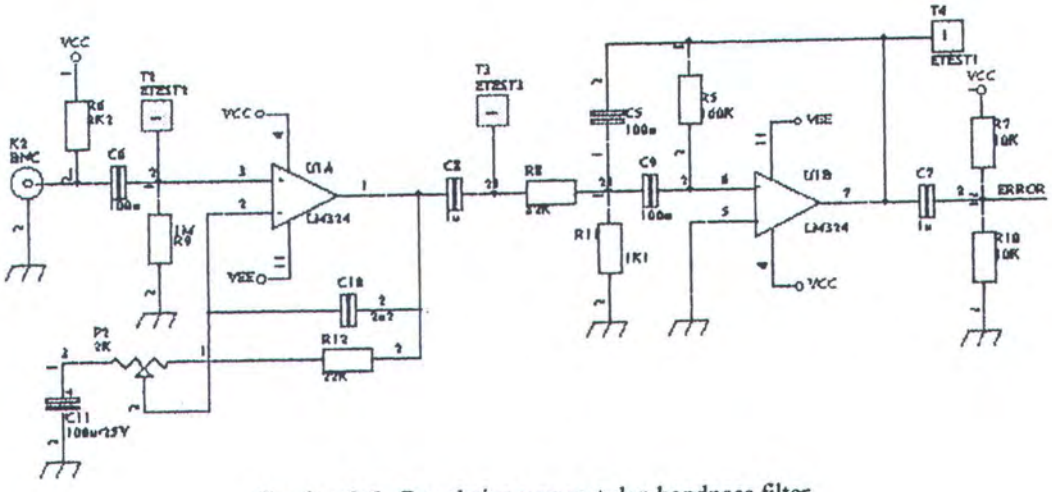
Seperti terlihat pada gambar rangkaian, Fuzzy logic controller AL220 yang terletak di sebelah kanan berfungsi untuk menghitung nilai RMS dari sinyal error yang masuk. Input yang digunakan hanya satu yaitu analog input 1 (AIN1) yaitu berupa sinyal error yang berasal dari sensor error setelah melalui rangkaian penguat dan bandpass filter. Input lainnya (AIN0, AIN2, dan AIN3) dihubungkan

ke VEE yaitu tegangan –5V. Pada umumnya input yang tidak digunakan dihubungkan ke ground, tetapi karena tegangan yang diberikan pada AL220 ialah 0V pada kaki PWDN dan –5V pada kaki GND. Hal ini dilakukan karena trankonduktansi amplifier menerima tegangan pengontrol antara –5V sampai 0V dan yang melakukan pengontrolan ini secara langsung ialah fuzzy logic controller AL220. Output yang digunakan juga hanya satu yaitu analog output 3 (AOUT3) berupa nilai RMS dari sinyal error (ERR PWR).

Fuzzy logic controller AL220 yang lain digunakan untuk mengontrol besarnya pergeseran fase dan besarnya penguatan dari sinyal anti noise. Input yang digunakan ialah analog input 1 (AIN1) berupa nilai RMS dari sinyal error (ERR PWR). Input lainnya (AIN0, AIN2, dan AIN3) dihubungkan ke VEE yaitu tegangan –5V. Outputnya ada dua yaitu sinyal level (AOUTA) pada analog output 1 (AOUT1) dan sinyal phase (AOUTB) pada analog output 3 (AOUT3). Sinyal level dihubungkan ke kaki pengatur dari tranconductance amplifier yang bertugas mengontrol besarnya penguatan (amplitudo) sinyal anti noise. Sinyal phase dihubungkan dihubungkan ke kaki pengatur dari tranconductance amplifier yang bertugas mengontrol besarnya pergeseran fase dari sinyal anti noise. Sinyal phase maupun sinyal level berupa teganan antara –5V sampai 0V.

3.2.5. Rangkaian Penguat dan Bandpass Filter untuk Sinyal Error

Gambar rangkaian penguat dan bandpass filter yang digunakan sebagai berikut:



Gambar 3.6. Rangkaian penguat dan bandpass filter

Rangkaian penguat diperlukan karena sinyal error adalah sinyal semakin lama semakin kecil amplitudonya.

Bandpass filter yang digunakan ialah bandpass dengan frekuensi tengah 120 Hz karena sinyal error merupakan hasil interferensi noise yang dihasilkan oleh kipas angin yang frekuensinya 120 Hz dengan sinyal anti noise yang dihasilkan oleh speaker dengan frekuensi yang sama. Faktor kualitas (Q) filter ialah 6. Sedangkan faktor penguatannya (G) sama dengan 1.

Harga komponen dapat dihitung dengan persamaan berikut ini:

$$Q = \frac{w_r}{BW}$$

$$BW = \frac{w_r}{Q} = \frac{2\pi \cdot 120}{6} = 125.6$$

$$R_1 = \frac{2}{BW * C} = \frac{2}{125,6 * 0,1\mu} = 159235 \approx 160K$$

$$R_4 = \frac{R_1}{4Q^2 - 2G} = \frac{160K}{4 * 6^2 - 2 * 1} = 1126,76 \approx 1K1$$

$$D_1 = \frac{R_1}{2G} = \frac{160K}{2 * 1} = 80K$$

Sinyal input dari bandpass filter ini berasal dari mikrofon 2 yang diletakkan di antara kipas angin dan speaker penghasil sinyal anti noise. Sinyal keluaran dari filter ini akan dimasukkan ke fuzzy logic controller AL220 untuk kemudian dihitung nilai RMS-nya.

3.3. PERANCANGAN PERANGKAT LUNAK

Perancangan perangkat lunak di sini ialah perangkat lunak untuk fuzzy logic controller AL220. Perangkat lunak untuk unit ini terdiri dari sekumpulan aturan yang digunakan FLC AL220 dalam menanggapi input-input yang diberikan. Langkah-langkah yang harus diperhatikan dalam membuat aturan fuzzy ini adalah:

1. Menentukan I/O yang diperlukan;
2. Membuat fuzzy variabel;
3. Membuat aturan fuzzy (fuzzy rules).

3.3.1. Penentuan Input/Output

Input Fuzzy logic controller AL220 yang digunakan untuk mengatur besarnya penguatan dan pergeseran fase sinyal anti noise ada empat buah, yaitu:

1. OnOff : Sinyal yang berfungsi untuk mengatur sistem aktif atau tidak
2. Error : Sinyal dari AL220 yang berfungsi menentukan nilai RMS dari error yang ada.
3. Phase : tidak digunakan
4. Y : tidak digunakan

Outputnya ada empat buah yaitu:

1. ErrDly : sinyal error yang telah didelay, untuk membandingkan harga error yang lama dengan harga error yang baru.
2. Level : sinyal keluaran untuk mengatur besarnya penguatan pada rangkaian transkonduktansi amplifier.
3. State : menyatakan nilai state dari sinyal masukan
4. PhaseInc : menyatakan besarnya pergeseran fase, dihubungkan ke transkonduktansi amplifier pengatur fase.

Input Fuzzy logic controller AL220 yang digunakan untuk menentukan besarnya nilai RMS dari sinyal error ada satu buah, yaitu:

1. Spare : tidak digunakan
2. ErrIn : Sinyal error dari error sensor yang akan ditentukan nilai RMS -nya.

Outputnya ada tiga buah yaitu:

1. Peak : menyatakan nilai puncak dari error
2. Acum : menyatakan nilai akumulasi dari error
3. Error : sinyal yang diumpan balikkan ke input
4. Ramp : output rms

3.3.2. Penentuan Fuzzy Variabel

Term yang digunakan masing-masing masukan AL220 yang digunakan untuk mengatur besarnya penguatan dan pergeseran fase ialah:

Error is High (64 , 0 , 1 , RI)

Error is Higher (ErrDly , 0 , 1 , LE)

Error is Low (64 , 0 , 1 , LI)

Error is LowSame (ErrDly , 0 , 1 , LI)

Error is Zero (10 , 0 , 1 , LI)

OnOff is Any (0 , 0 , 1 , RI)

OnOff is Off (128 , 0 , 1 , LI)

PhaseInc is HighOne (205 , 0 , 1 , RI)

PhaseInc is HighZero (205 , 0 , 1 , LI)

PhaseInc is LowOne (50 , 0 , 1 , RI)

PhaseInc is LowZero (50 , 0 , 1 , LI)

State is X1 (10 , 0 , 1 , SI)

State is X2 (20 , 0 , 1 , SI)

State is X3 (30 , 0 , 1 , SI)

State is X3Y1 (45 , 0 , 1 , SI)

State is X4 (40 , 0 , 1 , SI)

State is Xall (20 , 20 , 1 , SI)

State is Y1 (50 , 0 , 1 , SI)

State is Y2 (60 , 5 , 1 , SI)

State is Y3 (70 , 0 , 1 , SI)

State is Y3X1 (85 , 0 , 1 , SI)

State is Y4 (80 , 5 , 1 , SI)

State is Zero (0 , 0 , 1 , SI)

Term yang digunakan masing-masing masukan AL220 yang digunakan untuk menghitung besarnya nilai RMS dari error ialah:

Errln is <128 (128 , 0 , 1 , LI)

Errln is <Accum (Accum , 0 , 1 , LI)

Errln is >128 (128 , 0 , 1 , RI)

Errln is >Accum (Accum , 0 , 1 , RI)

Peak is Large (215 , 0 , 1 , RI)

Peak is Medium (170 , 0 , 1 , RI)

Peak is Small (127 , 0 , 1 , RI)

Ramp is >Errln (Errln , 0 , 1 , RI)

Ramp is 128 (128 , 0 , 1 , SI)

Ramp is 255 (255 , 0 , 1 , SI)

Ramp is Not255 (255 , 0 , 1 , SE)

Ramp is Zero (0 , 0 , 1 , SI)

3.3.3. Penentuan Rule

Rule yang digunakan AL220 yang digunakan untuk mengatur besarnya penguatan dan pergeseran fase ialah sebagai berikut:

If OnOff is Off Then Level = 0

If State is Zero Then Level = 127

If Error is High and State is X2 Then Level + 5

If Error is Low and State is X2 Then Level + 1

If Error is High and State is X4 Then Level - 5

If Error is Low and State is X4 Then Level - 1

If Error is High and State is X3Y1 Then Level + 5

If Error is Low and State is X3Y1 Then Level + 1#

If State is Zero Then PhaseInc = 255

If State is Y2 Then PhaseInc = 160

If State is Y1 Then PhaseInc = 255

If State is Y4 Then PhaseInc = 0

If State is Y3 Then PhaseInc = 95

If State is Y3X1 Then PhaseInc = 255

If State is Xall Then PhaseInc = 160:

If OnOff is Off Then State = 0

If State is Zero Then State = 60

If Error is LowSame and State is X1 Then State = 20

If State is X2 Then State = 10

If Error is Higher and State is X1 Then State = 40

If Error is LowSame and State is X3 Then State = 40

If State is X4 Then State = 30

If Error is Higher and State is X3 Then State = 45

If State is X3Y1 Then State = 60

If Error is LowSame and State is Y1 Then State = 60

If State is Y2 Then State - 5

If Error is Higher and State is Y1 Then State = 80

If State is Y4 Then State - 5

If Error is LowSame and State is Y3 Then State = 80

If Error is Higher and State is Y3 Then State = 85

If State is Y3X1 Then State = 20:

If State is X2 Then ErrDly + 0

If State is X4 Then ErrDly + 0

If State is Y1 Then ErrDly + 0

If State is Y3 Then ErrDly + 0

If OnOff is Any Then ErrDly = Error:

JMP 0

Rule yang digunakan AL220 yang digunakan untuk menghitung besarnya nilai RMS dari error ialah sebagai berikut:

If Errln is <128 and Peak is Large Then Accum - 1

If Errln is <128 and Peak is Medium Then Accum - 1

If Errln is <128 and Peak is Small Then Accum - 3

If Errln is >Accum Then Accum = Errln#

If Ramp is Zero Then Ramp = 255

If Errln is <128 and Ramp is 255 Then Ramp = 255

If Errln is <128 and Ramp is >Errln Then Ramp = Errln

If Errln is >128 and Ramp is 255 Then Ramp = Errln

If Errln is <128 and Ramp is Not255 Then Ramp - 20:

If Ramp is Zero Then Error = Accum:

If Errln is >128 and Errln is >Accum Then Peak = Errln#

JMP 0

BAB IV

PENGUJIAN SISTEM

Bab ini berisi hasil pengukuran dan pengujian terhadap sistem yang dibuat. Pengujian perangkat keras yang dilakukan terdiri dari pengujian bagian band pass filter dan pengujian bagian amplifier. Pengujian perangkat lunak yang dilakukan terdiri dari perangkat lunak untuk *fuzzy logic controller* AL220 yang berfungsi untuk menentukan besarnya pergeseran fase dan amplitudo dari sinyal input dan perangkat lunak untuk *fuzzy logic controller* yang berfungsi menentukan nilai RMS dari sinyal error.

4.1. PENGUJIAN BAGIAN BAND PASS FILTER

Pada bagian band pass filter 120Hz dengan faktor kualitas (Q) = 5, dan faktor penguatan (gain) = 10 dilakukan pengujian dan pengukuran yaitu dengan cara memberikan tegangan input beramplitudo tetap yaitu 1 Volt peak to peak, tetapi frekuensinya diubah-ubah. Outputnya diukur dengan osciloscope.

Hasil pengkuran bagian bandpass filter ini diperlihatkan pada tabel 4.1

Tabel 4.1 Hasil Pengukuran Bagian bandpass filter

No	Frekuensi (Hz)	Output analog (Volt)
1	40	0,1
2	60	0,9
3	80	3,6
4	100	8,4
5	120	9,8
6	140	8,7
7	160	6,6
8	180	2,1
9	200	0,5
10	220	0,2

4.2. PENGUJIAN BAGIAN AMPLIFIER

Pada bagian Amplifier pengujian dan pengukuran dilakukan dengan cara memberikan tegangan input pada amplifier diukur dengan osciloscop kemudian dibandingkan dengan output digital yang dihasilkan oleh ADC. Perhitungan error yang terjadi dalam pengukuran dapat dilakukan dengan cara sebagai berikut:

$$error(\%) = \frac{hasil\_perhitungan - hasil\_pengukuran}{hasil\_perhitungan} \bullet 100$$

Hasil pengkuran bagian amplifier yang menggunakan ini diperlihatkan pada tabel 4.2. Perhitungan error dari pengukuran dilakukan dengan persamaan diatas.

Tabel 4.2 Hasil Pengukuran Bagian Amplifier dengan gain = 8

No	Input(analog) (Volt)	Output(analog) (Volt)	Error(%)
1	0	0	0
2	0,02	0,15	6,25
3	0,05	0,42	5
4	0,08	0,62	3,125
5	0,1	0,77	3,75

6	0,12	0,95	1,04
7	0,14	1,1	2,68
8	0,15	1,18	1,67

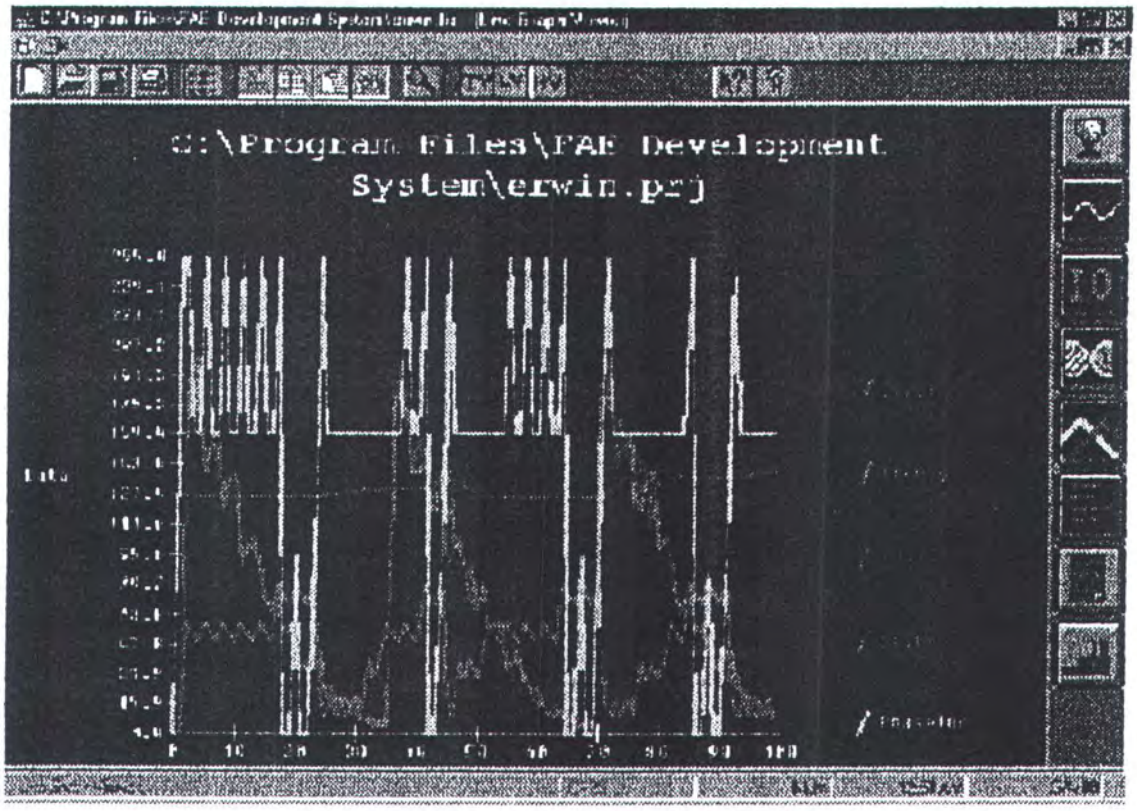
4.3. PENGUJIAN PERANGKAT LUNAK

Pengujian terhadap perangkat lunak dilakukan secara simulasi pada program FAE Development System.

4.3.1. Simulasi perangkat lunak AL220 yang berfungsi menentukan besarnya pergeseran fase dan penentu besarnya amplitudo.

Sinyal input yang digunakan ada dua yaitu OnOff dan Error. Sinyal OnOff berfungsi untuk mengaktifkan atau menonaktifkan sistem. Sinyal Error ialah sinyal umpan balik yang berfungsi untuk menentukan apakah error yang terjadi bertambah besar atau kecil. Sinyal outputnya ada dua yaitu Level dan PhaseInc. Sinyal level berfungsi mengatur besarnya amplitudo sinyal anti noise. Sinyal PhaseInc berfungsi mengatur besarnya pergeseran fase yang dilakukan terhadap sinyal noise.

Berikut ini adalah simulasi perangkat lunak *fuzzy logic controller* AL220 terhadap sinyal input yaitu sinyal error yang diberikan.

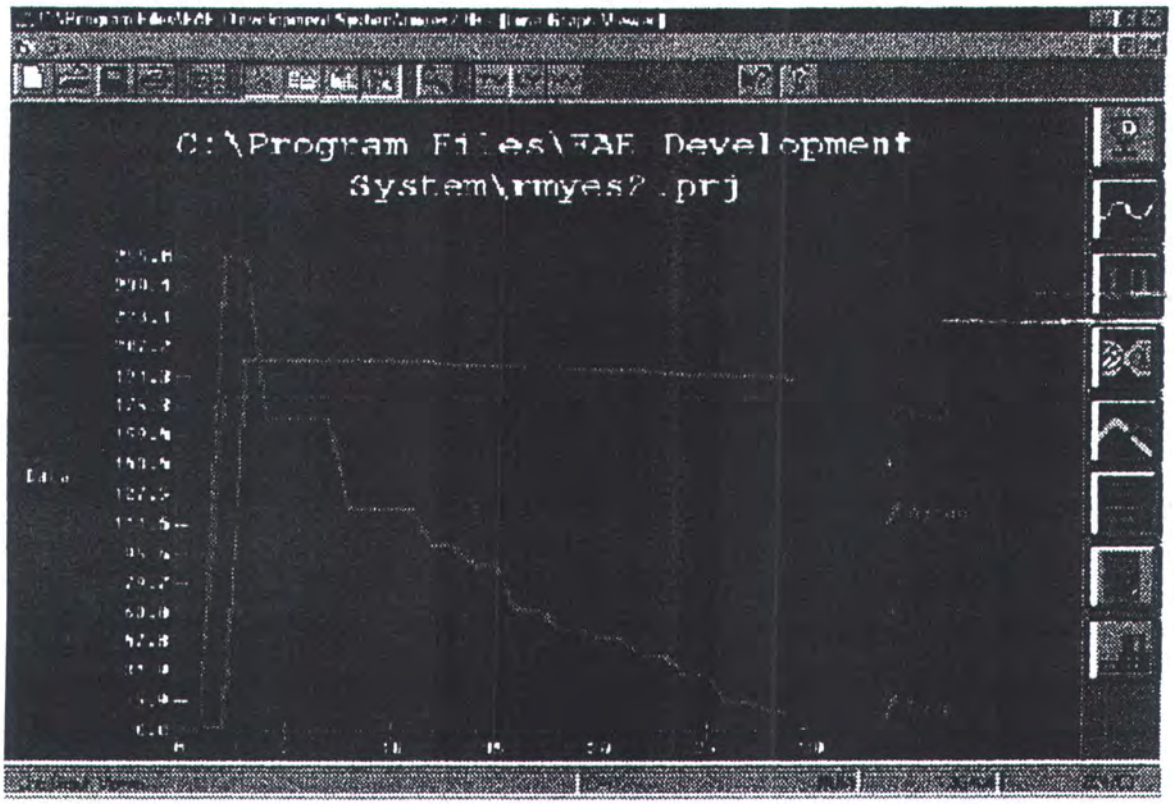


Gambar 4.1. Simulasi perangkat lunak AL220 pengontrol

4.3.2. Simulasi perangkat lunak AL220 yang berfungsi menentukan besarnya nilai RMS error.

Sinyal input yang digunakan ada satu yaitu Error.. Sinyal Error ialah sinyal yang berasal dari sensor error yang diletakkan di antara sumber noise dan sumber sinyal anti noise. Sinyal outputnya yaitu sinyal Ramp yang merupakan nilai RMS dari sinyal error.

Berikut ini adalah simulasi perangkat lunak *fuzzy logic controller* AL220 terhadap sinyal input yaitu sinyal error yang diberikan.



Gambar 4.2. Simulasi perangkat lunak AL220 RMS

4.4. PENGUJIAN BAGIAN *FUZZY LOGIC CONTROLLER AL220*

Pengujian *fuzzy logic controller* AL220 terdiri dari dua bagian, yaitu AL220 yang berfungsi menentukan nilai RMS dan AL220 yang berfungsi untuk mengontrol besarnya pergeseran fase dan amplitudo sinyal anti noise.

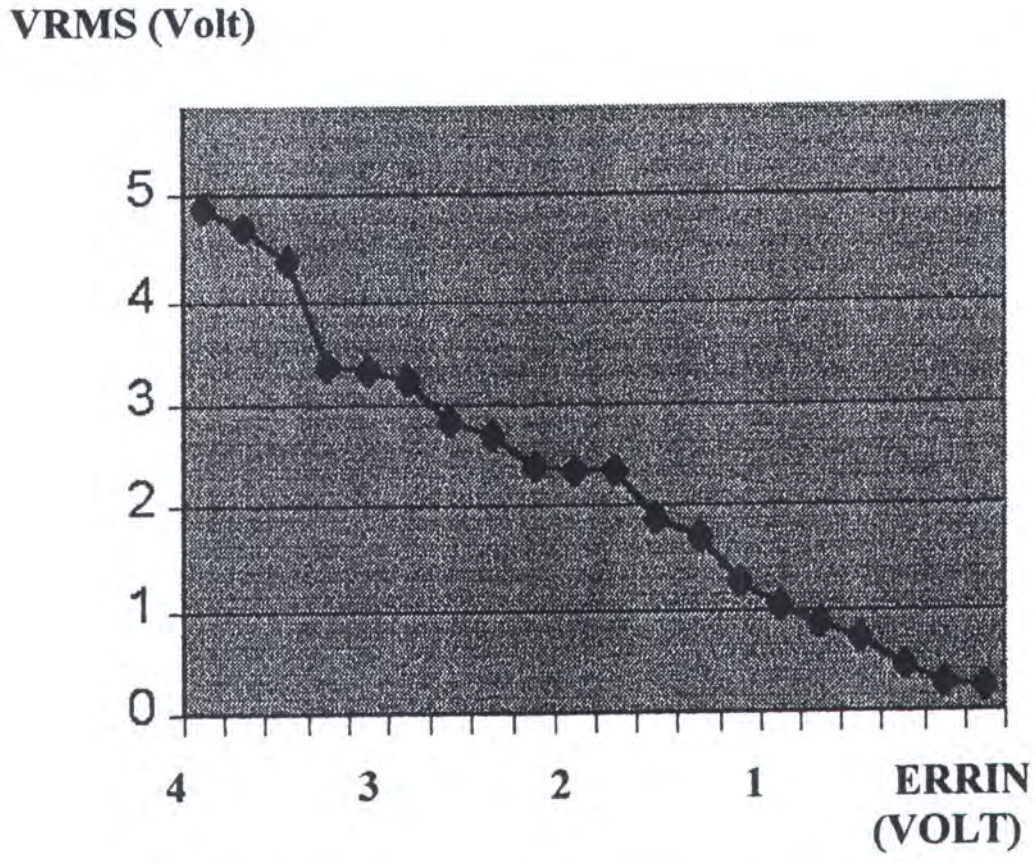
Pada masing-masing AL220 diberikan masukan tegangan tertentu dan pada bagian keluaran tegangannya diukur dengan voltmeter digital. Tabel 4.3. merupakan hasil pengujian respon *fuzzy logic controller* AL220 yang berfungsi menentukan nilai RMS terhadap perubahan masukan sedangkan tabel 4.4. merupakan hasil pengujian respon *fuzzy logic controller* AL220 yang berfungsi untuk mengontrol besarnya pergeseran fase dan amplitudo sinyal anti noise.

Tabel 4.3 Pengujian respon AL220 penentu nilai RMS terhadap masukan

MASUKAN		KELUARAN	
ERRIN (VOLT)	DATA	RMS (VOLT)	DATA
4	204	4,88	249
3,8	194	4,71	240
3,4	173	4,39	224
3,2	163	3,37	172
3,1	158	3,31	169
2,8	143	3,24	165
2,6	133	2,84	145

2,5	128	2,69	137
2,2	112	2,37	121
2,0	102	2,35	120
1,9	97	2,31	118
1,5	77	1,88	96
1,2	61	1,71	87
1,0	51	1,25	64
0,9	46	1,02	52
0,7	36	0,86	44
0,6	31	0,71	36
0,4	20	0,47	24
0,2	10	0,27	14
0,1	5	0,22	11

Data di atas ditampilkan dalam bentuk grafik seperti pada gambar 4.3. berikut ini



Gambar 4.3. Grafik pengujian respon AL220 penentu nilai RMS

Tabel 4.4 Pengujian respon AL220 penentu besarnya pergeseran fase dan amplitudo sinyal anti noise

MASUKAN				KELUARAN			
ONOFF		ERROR		LEVEL		PHASEINC	
VOLT	DATA	VOLT	DATA	VOLT	DATA	VOLT	DATA
0	0	3,5	179	0	0	0	0
0	0	3,1	158	0	0	0	0
0	0	2,5	128	0	0	0	0
0	0	2	102	0	0	0	0
0	0	1,4	71	0	0	0	0
0	0	1,1	56	0	0	0	0
0	0	0,6	31	0	0	0	0
5	255	2,8	143	2,43	124	4,52	231
5	255	2,5	128	2,35	120	4,18	245

5	255	2,3	117	2,31	118	2,82	144
5	255	2	102	2,57	131	3,59	183
5	255	1,9	97	2,33	119	4,.25	217
5	255	1,6	82	2,65	135	3,72	190
5	255	1,4	70	2,26	115	2,81	143
5	255	1,3	66	2,18	111	2,16	110
5	255	1,1	56	2,43	124	1,75	89
5	255	0,8	41	2,62	131	3,87	197
5	255	0,6	31	2,52	129	3,24	165
5	255	0,3	15	2,67	136	4,41	225
5	255	0,1	5	2,45	125	1,32	67

BAB V

PENUTUP

Bab ini merupakan penutup dari semua bab yang ada dan berisi tentang kesimpulan yang dapat diambil dari penelituan yang dilakukan dan berisi saran-saran untuk penelitian lebih lanjut.

5.1. KESIMPULAN

Dalam tugas akhir ini dapat diambil beberapa kesimpulan, antara lain:

1. Logika fuzzy dapat digunakan dalam menyelesaikan permasalahan pengurangan atau penghilangan noise dalam sistem fisik.
2. Unjuk kerja Fuzzy Logic Controller AL220 cukup memadai untuk digunakan dalam proses pengurangan noise.
3. Karena AL220 tidak diprogram seperti pada Digital Signal Processing (DSP) atau mikroprosesor konvensional, waktu prosesnya menjadi berkurang. Hal ini sesuai untuk sistem yang membutuhkan waktu real.
4. Sinyal error yang dihasilkan tetap besar karena pengaruh bunyi pantul pada dinding ruang.
5. Sinyal anti noise yang dihasilkan oleh sistem bercampur dengan noise yang dihasilkan oleh catu daya dan penguat pada rangkaian akibatnya bunyi yang terdengar bertambah keras.
6. Penempatan posisi mikrofon sebagai sensor error berpengaruh terhadap sinyal hasil interferensi yang ditangkapnya. Idealnya mikrofon ditempatkan di titik di mana hasil interferensi sinyal noise dan sinyal anti noise saling melemahkan, di antaranya yaitu posisi di tengah-tengah antara sumber noise dan sumber sinyal anti noise

5.2. SARAN

1. Sistem ini dapat dikembangkan pada permasalahan pengurangan noise yang lain yang lebih kompleks dengan menggunakan Fuzzy Logic Controller jenis lain yang mempunyai unjuk kerja lebih baik.
2. Penggunaan mikrofon yang berkualitas baik sangat membantu dalam penangkapan sinyal noise dan sinyal error yang ada sehingga hasil yang diperoleh lebih baik.

DAFTAR PUSTAKA

1. BasiConcepts, **AL220 STAND ALONE FUZZY LOGIC CONTROLLER,** Data Sheet.
2. Coughlin, Robert F., Frederick F. Drescoll. 1992. **PENGUAT OPERASIONAL DAN RANGKAIAN TERPADU LINEAR**. Institut Teknologi Wentworth. Diterjemahkan oleh Herman Widodo Soemitro. Jakarta: Penerbit Airlangga.
3. Fredrick W. Hughes. 1981. **OP-AMP HANDBOOK**. USA: Prentice Hall
4. Mohammad Jamshidi, Nader Vadiee, Timothy J. Ross. 1993. **FUZZY LOGIC AND CONTROL**, PTR Prentice Hall, Englewood Cliffs.
5. National Semiconductor. 1988. **LINIER DATA BOOK 1**. USA: National Semiconductor
6. Yan, Jun; Ryan, Michael; Power, James. 1994. **USING FUZZY LOGIC**, United Kingdom : Prentice Hall International Ltd.

Lampiran A

```
;          C:\Program Files\FAE Development System\erwin.LST


ITEM          MAX       USED
Data Size    = 8
Memory Size  = 256       141 X 14
Stack Size   = 0
Max Inputs   = 4         4
Max Outputs  = 4         4
Max Variables = 0        0
PWM Circuits  = 0
Page Register =
Inter. Page  = False
Timer Size   = 0
# Rules                 37
CAW                     19


;
VARIABLES
;
;
INPUTS
;
OnOff = 0     32
Error = 0     33
Phase = 0     34
y = 0         35
;
OUTPUTS
;
ErrDly = 0    36
Level = 0     37
State = 0     38
PhaseInc = 0  39
;
COMPARISON FUNCTIONS
;
Error is High ( 64 , 0 , 1 , RI )
Error is Higher ( ErrDly , 0 , 1 , LE )
Error is Low ( 64 , 0 , 1 , LI )
Error is LowSame ( ErrDly , 0 , 1 , LI )
Error is Zero ( 10 , 0 , 1 , LI )
```

```
OnOff is Any ( 0 , 0 , 1 , RI )
OnOff is Off ( 128 , 0 , 1 , LI )
PhaseInc is HighOne ( 205 , 0 , 1 , RI )
PhaseInc is HighZero ( 205 , 0 , 1 , LI )
PhaseInc is LowOne ( 50 , 0 , 1 , RI )
PhaseInc is LowZero ( 50 , 0 , 1 , LI )
State is X1 ( 10 , 0 , 1 , SI )
State is X2 ( 20 , 0 , 1 , SI )
State is X3 ( 30 , 0 , 1 , SI )
State is X3Y1 ( 45 , 0 , 1 , SI )
State is X4 ( 40 , 0 , 1 , SI )
State is Xall ( 20 , 20 , 1 , SI )
State is Y1 ( 50 , 0 , 1 , SI )
State is Y2 ( 60 , 5 , 1 , SI )
State is Y3 ( 70 , 0 , 1 , SI )
State is Y3X1 ( 85 , 0 , 1 , SI )
State is Y4 ( 80 , 5 , 1 , SI )
State is Zero ( 0 , 0 , 1 , SI )
;
;
```

PROGRAM STORE CONTENTS

Adrs	Data	Rule
0	1A0	If OnOff is Off Then Level = 0
1	40	
2	1081	
3	A6	If State is Zero Then Level = 127
4	41	
5	1082	
6	121	If Error is High and State is X2 Then Level + 5
7	43	
8	A6	
9	44	
A	1105	
B	1A1	If Error is Low and State is X2 Then Level + 1
C	43	
D	A6	
E	44	
F	1106	
10	121	If Error is High and State is X4 Then Level - 5
11	43	
12	A6	
13	47	
14	1185	

```
15      1A1     If Error is Low and State is X4 Then Level - 1
16      43
17      A6
18      47
19      1186
1A      121     If Error is High and State is X3Y1 Then Level + 5
1B      43
1C      A6
1D      48
1E      1105
1F      1A1     If Error is Low and State is X3Y1 Then Level + 1#
20      43
21      A6
22      48
23      1106
24      2125
25      A6      If State is Zero Then PhaseInc = 255
26      41
27      1089
28      A6      If State is Y2 Then PhaseInc = 160
29      14A
2A      108B
2B      A6      If State is Y1 Then PhaseInc = 255
2C      4C
2D      1089
2E      A6      If State is Y4 Then PhaseInc = 0
2F      14D
30      1081
31      A6      If State is Y3 Then PhaseInc = 95
32      4E
33      108F
34      A6      If State is Y3X1 Then PhaseInc = 255
35      50
36      1089
37      A6      If State is Xall Then PhaseInc = 160:
38      104
39      108B
3A      2027
3B      1A0     If OnOff is Off Then State = 0
3C      40
3D      1081
3E      A6      If State is Zero Then State = 60
3F      41
40      108A
41      1A1     If Error is LowSame and State is X1 Then State = 20
```

```
42      1064
43      A6
44      51
45      1084
46      A6      If State is X2 Then State = 10
47      44
48      1091
49      1E1     If Error is Higher and State is X1 Then State = 40
4A      1064
4B      A6
4C      51
4D      1087
4E      1A1     If Error is LowSame and State is X3 Then State = 40
4F      1064
50      A6
51      52
52      1087
53      A6      If State is X4 Then State = 30
54      47
55      1092
56      1E1     If Error is Higher and State is X3 Then State = 45
57      1064
58      A6
59      52
5A      1088
5B      A6      If State is X3Y1 Then State = 60
5C      48
5D      108A
5E      1A1     If Error is LowSame and State is Y1 Then State = 60
5F      1064
60      A6
61      4C
62      108A
63      A6      If State is Y2 Then State - 5
64      14A
65      1185
66      1E1     If Error is Higher and State is Y1 Then State = 80
67      1064
68      A6
69      4C
6A      108D
6B      A6      If State is Y4 Then State - 5
6C      14D
6D      1185
6E      1A1     If Error is LowSame and State is Y3 Then State = 80
```

6F	1064	
70	A6	
71	4E	
72	108D	
73	1E1	If Error is Higher and State is Y3 Then State = 85
74	1064	
75	A6	
76	4E	
77	1090	
78	A6	If State is Y3X1 Then State = 20:
79	50	
7A	1084	
7B	2026	
7C	A6	If State is X2 Then ErrDly + 0
7D	44	
7E	1101	
7F	A6	If State is X4 Then ErrDly + 0
80	47	
81	1101	
82	A6	If State is Y1 Then ErrDly + 0
83	4C	
84	1101	
85	A6	If State is Y3 Then ErrDly + 0
86	4E	
87	1101	
88	120	If OnOff is Any Then ErrDly = Error:
89	41	
8A	10E1	
8B	2024	
8C	3000	JMP 0

TABLE CONTENTS

REGISTER INITIALIZATION CONTENTS

120	0
121	0
122	0
123	0
124	0
125	0
126	0

127 0

CENTER, ACTION, WIDTH CONTENTS

140	80
141	0
142	7F
143	40
144	14
145	5
146	1
147	28
148	2D
149	FF
14A	3C
14B	A0
14C	32
14D	50
14E	46
14F	5F
150	55
151	A
152	1E

; C:\Program Files\FAE Development System\rmyes2.LST

ITEM	MAX	USED
Data Size	= 8	
Memory Size	= 255	54 X 14
Stack Size	= 4	
Max Inputs	= 4	2
Max Outputs	= 15	4
Max Variables	= 32	0
PWM Circuits	= 0	
Page Register	=	
Inter. Page	= False	
Timer Size	= 0	
# Rules		12
CAW		9

```
;
VARIABLES
;
;
INPUTS
;
Spare = 2     32
Errln = 2     33
;
OUTPUTS
;
Peak = 0      34
Accum = 0     35
Error = 0     36
Ramp = 0      37
CONTROL = 0   61
INTERRUPT = 0  62
PRESCALE = 0   63
;
COMPARISON FUNCTIONS
;
Errln is <128 ( 128 , 0 , 1 , LI )
Errln is <Accum ( Accum , 0 , 1 , LI )
Errln is >128 ( 128 , 0 , 1 , RI )
Errln is >Accum ( Accum , 0 , 1 , RI )
Peak is Large ( 215 , 0 , 1 , RI )
Peak is Medium ( 170 , 0 , 1 , RI )
Peak is Small ( 127 , 0 , 1 , RI )
Ramp is >Errln ( Errln , 0 , 1 , RI )
Ramp is 128 ( 128 , 0 , 1 , SI )
Ramp is 255 ( 255 , 0 , 1 , SI )
Ramp is Not255 ( 255 , 0 , 1 , SE )
Ramp is Zero ( 0 , 0 , 1 , SI )
;
;
   PROGRAM STORE CONTENTS

Adrs      Data      Rule

0         1A1       If Errln is <128 and Peak is Large Then Accum - 1
1         40
2         124
3         42
4         1183
5         1A1       If Errln is <128 and Peak is Medium Then Accum - 1
```

```
6       40
7       124
8       44
9       1183
A       1A1     If Errln is <128 and Peak is Small Then Accum - 3
B       40
C       124
D       45
E       1186
F       121     If Errln is >Accum Then Accum = Errln#
10      1065
11      10E1
12      2125
13      A7      If Ramp is Zero Then Ramp = 255
14      41
15      1087
16      1A1     If Errln is <128 and Ramp is 255 Then Ramp = 255
17      40
18      A7
19      47
1A      1087
1B      1A1     If Errln is <128 and Ramp is >Errln Then Ramp = Errln
1C      40
1D      127
1E      1061
1F      10E1
20      121     If Errln is >128 and Ramp is 255 Then Ramp = Errln
21      40
22      A7
23      47
24      10E1
25      1A1     If Errln is <128 and Ramp is Not255 Then Ramp - 20:
26      40
27      E7
28      47
29      1188
2A      2027
2B      A7      If Ramp is Zero Then Error = Accum:
2C      41
2D      10E5
2E      2026
2F      121     If Errln is >128 and Errln is >Accum Then Peak = Errln#
30      40
31      121
32      1065
```

```
33      10E1
34      2124
35      3000    JMP 0
```

TABLE CONTENTS

REGISTER INITIALIZATION CONTENTS

```
11F     2
120     2
123     0
124     0
125     0
126     0
```

CENTER, ACTION, WIDTH CONTENTS

```
13F     80
140     0
141     D7
142     1
143     AA
144     7F
145     3
146     FF
147     14
```

BasiConcepts

AL220, Stand-Alone Fuzzy Logic Controller

Features

- Complete, Single-Chip Fuzzy Logic Processor
- Flexible, Self-Adapting Control
- EEPROM Storage
- Four 8-Bit Analog Inputs
- Four 8-Bit Analog Outputs
- Six Types of Membership Functions
- 111 Fuzzy Variables
- Up to 50 Rules
- PC-Based Development System
- 18-Pin DIP or 20-Pin SOIC Package

Applications

- Smart Appliances
- Power and Battery Management
- Automotive
- Motor Control
- Communications

Description

The AL220 is an inexpensive, high-performance, stand-alone Fuzzy Logic Processor. The device performs Fuzzy Logic calculations directly in hardware. Because the AL220 is a dedicated controller, it offers superior ease of use, performance, features, and robustness in harsh operating environments.

Fuzzy Logic is a powerful processing methodology that easily accommodates imprecise input data and system nonlinearities for rapid development of robust control systems.

The methodology uses linguistic descriptions of systems, making it intuitive and simple to use. Fuzzy Logic can be used to inexpensively add intelligence to a wide variety of products to improve performance and features, and to increase efficiency.

Fuzzy Membership functions and rules are stored as parameters in EEPROM. The memory organization is flexible and efficiently adapts to the requirements of the application. The device stores 111 fuzzy variables, which are organized into as many as 50 rules.

The device supports six different types of membership functions to meet the requirements of any application. Membership functions are constant-slope and need only the type, width, and center specified.

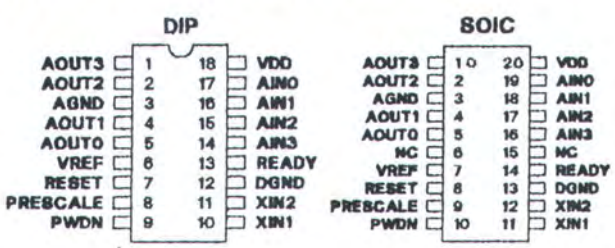
The AL220 offers Floating membership functions. The center and width of any membership function can be made to "Float" or vary dynamically. Floating membership functions can be used to measure derivatives, build timers, or adjust to drift in sensors.

Two methods of defuzzification are available: immediate and accumulate. The immediate mode drives an output with a specific value. The accumulate mode adds to the output's previous value.

Applications information is easily entered using the FL1000 Applications Expert development system running under Windows. Little knowledge of Fuzzy Logic is needed to use the devices or the system.

The AL220 is available in a 18-pin DIP or 20-pin SOIC package. The devices are ideal for a wide range of applications including appliances, motor control, automotive, and industrial systems.

Figure 1. Package Pin Assignments



312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts Web Site: www.BasiConcepts.com

n Description

uts

SET An active-low signal that initializes the device. ET should remain active for at least eight clock es to ensure proper operation. RESET can be driven power-up delayed circuit. Asserting RESET during ation causes the AL220 to enter its low-power mode.

(3:0) Analog input data. Analog data is internally erted to 8-bit digital data. Unused inputs should be ected to ground.

Clock input, driven by an external clock.

ESCALE A logic level one puts the device into cale mode while a zero causes normal operation. The an be grounded if prescale mode is never used. After ET is deasserted, the PRESCALE pin must be held logic low for at least four clock cycles. Prescale ation is described later in this document.

puts

UT(3:0) Analog output data. Eight-bit digital data is nally converted to an analog level.

ADY After a reset, this pin signals that the device is ng to sample and process data.

F Filters the internal reference voltage. Connect to nd through a 0.1µF capacitor.

e 1. Absolute Maximum Ratings $T_A = 25^oC$

Parameter	Min	Max	Units
GND	-0.5	7.0	V
GND	0	0	V
Input	0	V_{DD}	V
g Input	0	V_{DD}	V
Dissipation		100	mW
ge Temperature	-50	150	°C

Table 2. Analog Conversion Specifications

Parameter	Value	Units
Resolution	1	BIT
Zero Code Error	1±	LSB
Full Scale Error	1±	LSB
Signal-to-Noise Ratio	45	dB min
Slew Rate, Tracking	1.6	V/mS max
Sampling Rate	10kHz	Per Channel

Table 3. Specifications and Recommended Operating Conditions

Parameter	Min	Norm	Max	Units
V_{DD} Supply Voltage	4.75	5.0	5.25	V
I_{DD} Supply Current				mA
I_{OL} Digital Output Low-Level Current			5	mA
I_{OH} Digital Output High-Level Current			-5	mA
F Clock Frequency	1		10	MHz
V_{IL} Digital Input Low-Level Voltage	0		0.8	V
V_{IH} Digital Input High-Level Voltage	3.5		V_{DD}	V
I_{IL} Digital Input Low-Level Current			-40	µA
I_{IH} Digital Input High-Level Current			1	µA
Z_{IN} Analog Input Impedance	100	150	250	kohm
V_{IN} Analog Input Voltage Range	0		V_{DD} -0.5	V
V_O Analog Output Voltage Range	V_{SS}+0.5		V_{DD}-0.5	V
I_O Analog Output Current	-5		5	mA
T_W Reset Pulse Width	100			mS
T_{RV} Reset Inactive Before Clock	10			mS
T_A Operating Ambient Temperature	0		70	°C

312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts Web Site: www.BasiConcepts.com

e AL220

device is a dedicated, stand-alone Fuzzy Logic roller. It performs all calculations in hardware and not require software. The device is configured for a icular application through simple rules.

vice Architecture

agram of the AL220 is shown in Figure 2. The main ents are the Fuzzifier, Defuzzifier, and controller. Fuzzifier converts input data into Fuzzy data. The zifier, in conjunction with the controller, evaluates zy data by a user-defined set of rules that describes the system is to be controlled. When the rules have evaluated, the Defuzzifier assigns an action value to appropriate output.

veloping a Fuzzy Logic System

understand the operation of the device, you need to erstand how to enter a Fuzzy Logic model and how to orm Fuzzy Logic calculations. The following sections ain the concepts underlying a Fuzzy system.

Membership Functions

Membership functions are separate categories used to define the range over which an input can vary. Membership functions are compared with input data to see where the data falls on them. They have names selected by the designer, such as Hot, Fast, or Tall, that classify data.

A household thermometer can be used to illustrate the concept of membership functions and show how Fuzzy Logic works like human reasoning. A person asked to divide the range of a thermometer according to comfort might designate temperatures as follows:

Below 60°F = Cold

60°F to 70°F = Cool

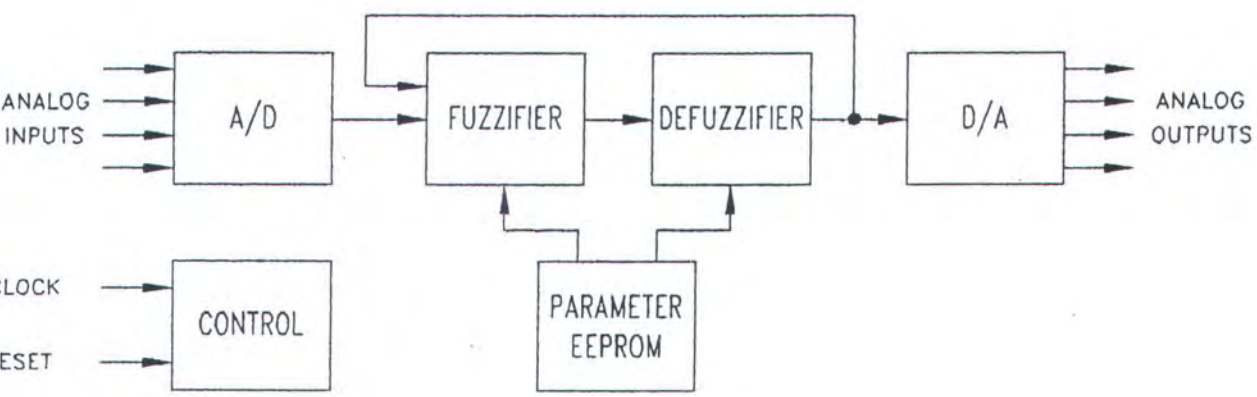
70°F to 75°F = Moderate

75°F to 85°F = Warm

Above 85°F = Hot

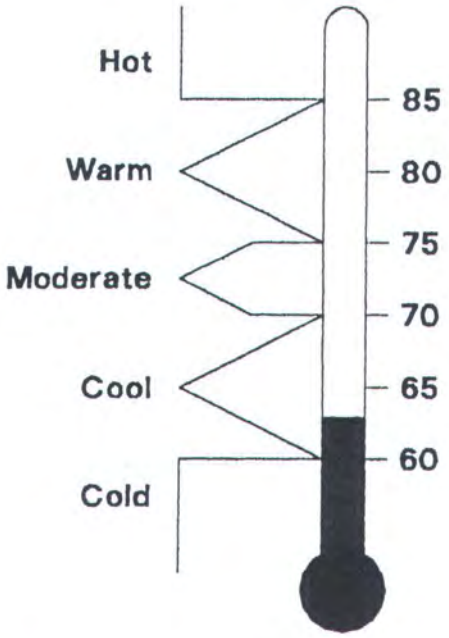
These divisions are an intuitive way for a person to consider temperature because they are based on the senses. A person could describe a room at 62°F as being cool. In Fuzzy Logic the five divisions are called membership functions and are represented in Figure 3. The membership functions can be separate, as shown, or they may overlap. That allows for data falling in the overlapped area to be a member of both functions. A temperature, for example, may be described as somewhat cool and somewhat cold.

ure 2. AL220 Block Diagram



312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts Web Site: www.BasiConcepts.com

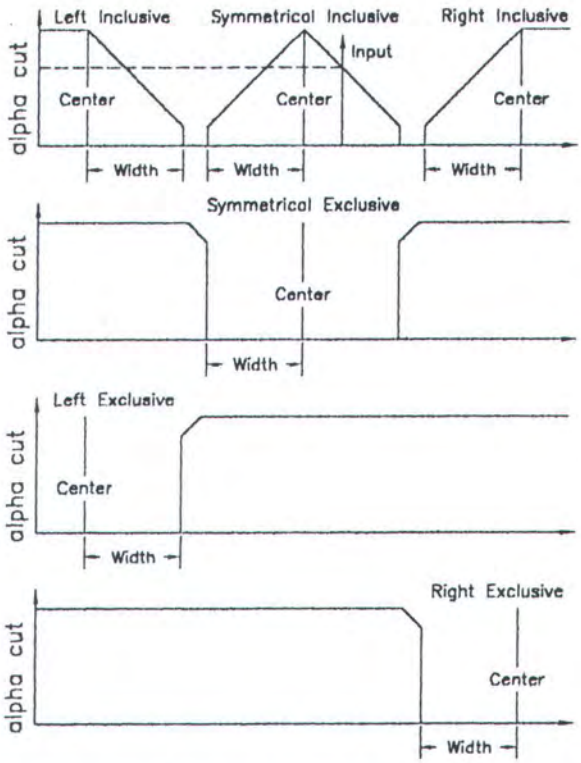
gure 3. Temperature Membership Functions



e AL220 supports six different, constant slope, mbership functions as shown in Figure 4. They consist Left, Symmetrical, and Right Inclusive functions and ir inverses, Exclusive functions. They are defined with ype that specifies a shape, with numerical values for center and width.

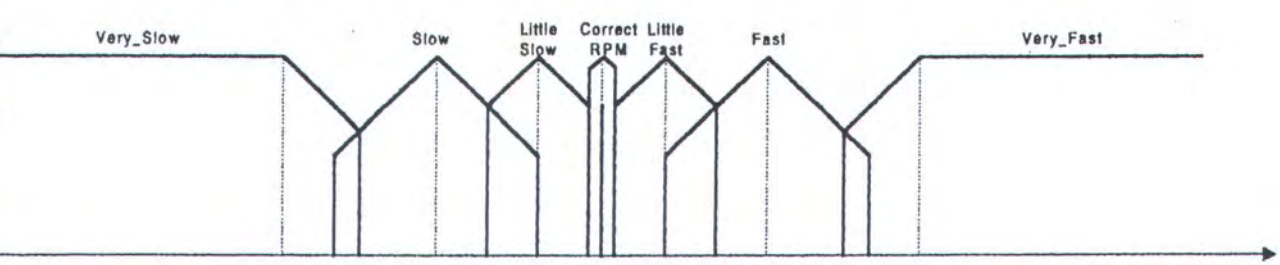
reful selection of membership functions simplifies the cription of many models. For example, single right or inclusive membership functions are used to cover ge ranges of values at the ends of the range of input iation. In the thermometer example, Cold would be resented by a left inclusive membership function and by a right inclusive membership function.

Figure 4. Supported Membership Function Types



Precise control about the desired operating point can be of the narrow symmetrical inclusive type, which provides more precise control. Many motor control applications require such precision. An example of a mixture of different types and widths of membership functions used to monitor the velocity of a motor is given in Figure 5.

ure 5. Velocity Membership Functions



312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

embership functions can be overlapped to create new apes such as trapezoids as shown in Figure 6. The apezoid is formed by overlaying a left and a right clusive membership function. Inputs falling into the apezoid are members of both functions.

gure 6. Overlapped Membership Functions

uzzy Variables

fuzzy variable is a linguistic expression representing e association of an input value against the membership nctions covering an axis. Fuzzy variables reference a embership function and an input variable. An example a fuzzy variable is as follows:

If Temperature is Cool

this example, 'Temperature' refers to an input and ool' to a membership function.

e association is performed by the Fuzzifier. The result fuzzy variable value which represents the degree to ich the input data matched the membership function. zzy data values are numerical and range from 0 to 63 the AL220. See Figure 7 for an example of a fuzzy iable evaluation.

Rules

A rule consists of one or more fuzzy variables and an action output value. Rules are used to tell the controller how to respond to changes in input data.

In the examples below, both rules contain two fuzzy variables. Rules are entered into the INSiGHT development system in the following format:

Output = -5 if Velocity is Fast and Acceleration is Positive

Output= +5 if Velocity is Little_Slow and Acceleration is Zero

In the first rule, the first fuzzy variable is the same as in the previous example and the second fuzzy variable is 'Acceleration is Positive.' The action '$\pm$5' is a numerical value that is applied to an output to slow down or speed up the motor.

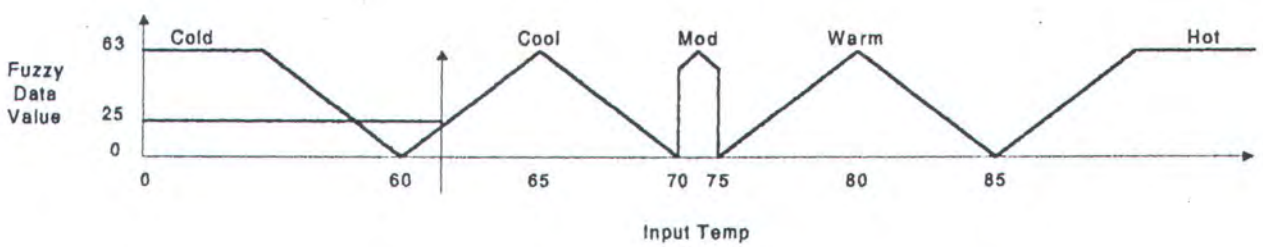
Rule Evaluation

There are several methods for evaluating Fuzzy Logic rules. The AL220 evaluates rules using the two-step MAX-of-MINs technique.

In the first step (MIN), all the values for the fuzzy variables in a rule are compared; the lowest value represents the rule. In the second step (MAX), the values for the rules are compared; the rule with the highest value wins.

The way membership functions, fuzzy variables, and rules are defined and organized depends on the requirements of the application. You need to understand the physical properties of the system to be controlled before entering the Fuzzy model. If you are armed with that knowledge, however, entering a model is a straightforward process.

gure 7. Fuzzification of Input Temperature



312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

embership functions can be overlapped to create new apes such as trapezoids as shown in Figure 6. The apezoid is formed by overlaying a left and a right clusive membership function. Inputs falling into the apezoid are members of both functions.

gure 6. Overlapped Membership Functions

uzzy Variables

fuzzy variable is a linguistic expression representing e association of an input value against the membership nctions covering an axis. Fuzzy variables reference a embership function and an input variable. An example a fuzzy variable is as follows:

If Temperature is Cool

this example, 'Temperature' refers to an input and ool' to a membership function.

e association is performed by the Fuzzifier. The result fuzzy variable value which represents the degree to ich the input data matched the membership function. zzy data values are numerical and range from 0 to 63 the AL220. See Figure 7 for an example of a fuzzy iable evaluation.

Rules

A rule consists of one or more fuzzy variables and an action output value. Rules are used to tell the controller how to respond to changes in input data.

In the examples below, both rules contain two fuzzy variables. Rules are entered into the INSiGHT development system in the following format:

Output = -5 if Velocity is Fast and Acceleration is Positive

Output= +5 if Velocity is Little_Slow and Acceleration is Zero

In the first rule, the first fuzzy variable is the same as in the previous example and the second fuzzy variable is 'Acceleration is Positive.' The action '$\pm$5' is a numerical value that is applied to an output to slow down or speed up the motor.

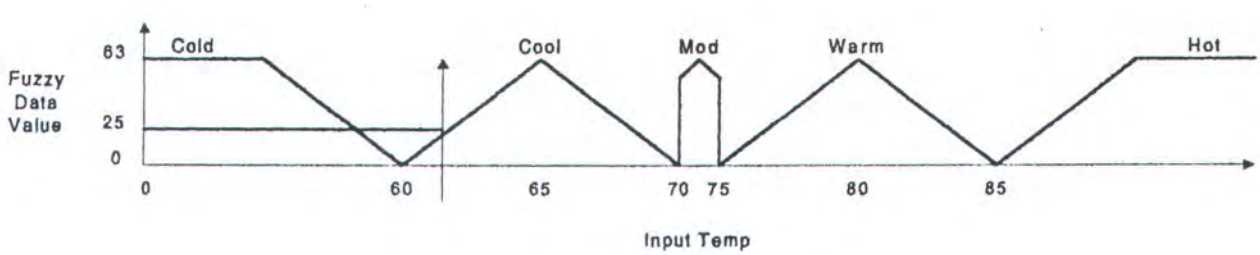
Rule Evaluation

There are several methods for evaluating Fuzzy Logic rules. The AL220 evaluates rules using the two-step MAX-of-MINs technique.

In the first step (MIN), all the values for the fuzzy variables in a rule are compared; the lowest value represents the rule. In the second step (MAX), the values for the rules are compared; the rule with the highest value wins.

The way membership functions, fuzzy variables, and rules are defined and organized depends on the requirements of the application. You need to understand the physical properties of the system to be controlled before entering the Fuzzy model. If you are armed with that knowledge, however, entering a model is a straightforward process.

gure 7. Fuzzification of Input Temperature

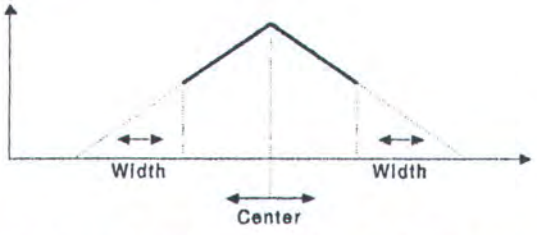


312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

loating Membership Function

unique feature of the AL220 is the Floating embership function. As shown in Figure 8, floating embership functions have center and width values that ry dynamically. In ordinary membership functions the nter and width are fixed values stored in memory. In a ating membership function, these values come from y of the inputs or outputs.

gure 8. Floating Membership Functions



y membership can be specified as floating during ign entry. The floating membership function changes center or width value as data from a selected input or put changes.

r example, two fuzzy variables with their membership ctions described parenthetically could be defined nventionally and used in a rule as follows:

IN1 is small (0, 25, Symmetrical Inclusive)

IN2 is small (0, 25, Symmetrical Inclusive)

ere the first number, zero, refers to the center and the ond, 25, to the width of the membership function.

Output = +1 if IN1 is small and IN2 is small

ere the fuzzy variable 'IN1 is small' compares input l against the conventional membership function all.'

ating membership functions make the same cription more concisely in the following fuzzy iable and rule:

is small_difference (IN2, 25, Symmetrical Exclusive)

put = +1 if IN1 is small_difference

the fuzzy variable, the center of the membership ction small_difference is defined by the value of IN2 ed in the input's latch.

Fuzzification, an input is subtracted from the center of mbership functions and the result inverted to measure v closely it matches the center value. When the device zifies a membership function with a floating center, it tracts one input from another.

Floating membership functions allow you to use a fuzzy variable that directly measures the difference between two inputs. The technique can be used, as in the example, to calibrate changes in a sensor over time.

The sensor's quiescent value is compared to a set voltage. Calibration rules check the degree of mismatch and store a correction value in an output. If the inputs are in calibration, the centers will match and the correction value is zero. Large mismatches will store large corrections.

The correction is used to adjust the floating center of a membership function in rules that process sensed data.

Floating membership functions can also be used to obtain the derivative of an input value by means similar to those described in the example above. A rule can reference an input as a floating action value causing it to be passed directly to an output.

During the next sample of the input, the output value selects the membership function center value, which has the effect of subtracting the previous input value from the current value. The difference can be referenced by a fuzzy variable in a rule.

An example of the use of an input/action value would be in measuring the acceleration of a motor. A rule that stores an input value into an output could be written as follows:

VALUE_TO - IN1 IF IN1 IS MUST_WIN (0, 0, Right Inclusive)

The rule references IN1 as an action value. The membership function MUST_WIN is a Right Inclusive type that begins at zero so that, regardless of the value of IN1, the rule must win and the value of IN1 is stored in the output.

A second rule calculates the derivative and adjusts the output that drives the motor.

ACCEL ± if IN1 is VALUE_T1 (VALUE_TO, 25, Symmetrical Inclusive)

The rule determines whether the input's value at T1 is within 25 of its value at T0. In an actual application, there would be other membership functions to determine the polarity of the derivative and other rules to cover larger adjustments to larger variations.

The above examples of floating membership functions are straightforward. In an actual application, floating membership functions can be used extensively to save memory because they use fewer fuzzy variables and rules to detect differences between inputs than conventional functions do.

312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

evice Operation

ocessing data involves several steps. First, sampled alog data is converted to digital data. Next, the zzifier compares the digital input data with the fuzzy riables to find a value for the fuzzy variable. The zzifier also performs the MAX-of-MIN calculation to termine the wining rule. Finally, the Defuzzifier termines the winning rule's action value and sends it conversion to an analog output or internal feedback.

uzzfier

e Fuzzifier compares input data with membership nctions to calculate a Fuzzy variable value. When the IN calculation has been performed on all the fuzzy riables in a rule, the value representing the rule is ored. When the MAX calculation has been performed all the rules referencing an output, the winning rule's ion value is passed to the Defuzzifier.

les are evaluated in the order they are entered. Any e can reference any output. Outputs can be referenced peatedly in a rule set.

nen a rule or group of rules affecting an output has en entered and the next rule entered references another tput, the Last Rule causes the output to be updated with action value of the winning rule.

after processing rules that affect other outputs, the ocessor encounters another rule or group of rules erencing the same output, then it will update the tput again. An output then may be updated as many nes during a processing cycle as there are separate oups of rules referencing it.

mentioned previously, input sampling is continuous. alog output values are also updated continuously. ring the course of a processing cycle, a fuzzy variable y use a data sample from the previous sample cycle or m the current cycle depending on where the input mple cycle is relative to the processing cycle. Should re than one group of rules reference the same input l output, then the output value may change more than ce during a processing cycle based on different input a.

he order in which these outputs change, or the number times an output changes during a processing cycle is nificant, then the order of rules should be considered en they are entered.

fuzzifier

winning rule's action value and mode data are sed to the Defuzzifier block. Digital data from the Defuzzifier is latched and converted to analog to drive the outputs or looped back internally.

If all the rules in a group referencing an output evaluate to zero, then the output will not change its value. If more than one rule evaluates to the same highest nonzero value, then the first of those rules entered will win and its action will determine the output.

Defuzzification Methods

Defuzzification causes the action value of the winning rule to drive an output. The device supports two methods of defuzzification, immediate and accumulate. Either of the two modes depicted in Figures 9 and 10 can be selected for a rule.

Immediate Mode functions like a lookup table, where the action value assigned to the winning rule during entry is applied to an output.

Immediate defuzzification is useful when the output value must be absolute or when large changes are required.

Accumulate Mode increments or decrements the existing output by the action value for the winning rule. The output is a function of the current action and the previous output.

Accumulate defuzzification can be used for subtle changes to outputs when the system under control is near a desired operating point. It is also useful for timing functions.

Figure 9. Immediate Defuzzification

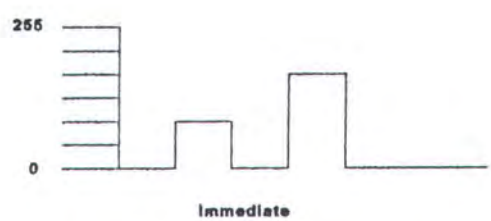
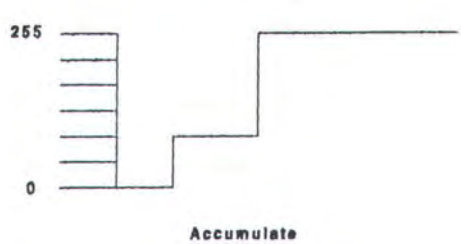


Figure 10. Accumulate Defuzzication



312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

emory Organization

AL220 contains a 256-byte EEPROM for lications parameter storage. The last 32 bytes store d memberhip function Center and Width values. The aining 224 bytes are organized as one or more rules one or more fuzzy variables per rule.

h rule requires two bytes, plus an additional two bytes each fuzzy variable in that rule. A rule containing fuzzy variables, for example, would use 12 bytes.

memory is organized into three sections, defined as e/Fuzzy Variable storage, Center storage, and Width age. Memory organization is illustrated in Table 4.

ble 4. Memory Organization

Address	Hex Address	Function
0	00	Rules
<>	<>	<>
223	DF	Rules
224	E0	Centers
<>	<>	<>
239	EF	Centers
240	F0	Widths
<>	<>	<>
255	FF	Widths

e and fuzzy Variable Storage

es are organized as groups of one or more fuzzy ables. Each fuzzy variable is made up of two bytes, as ribed in Tables 5 and 6. The first byte is stored at addresses and the second at odd addresses.

types are divided into fields that control how data is essed. The three least significant bits of the even byte ne either the membership functions type or whether previous fuzzy variable was the last of the rule or the fuzzy variable of the last rule referencing the output.

n the least significant field selects a membership tion type, the five most significant bits select the t source for the fuzzy variable. The five-bit field is ivided into a three-bit field that selects the input ce from one of the four input pins or output latches. remaining two most significant bits define whether center and width of the membership function are ing or fixed.

The Type code Last Fuzzy variable (001) signals that the last fuzzy variable of the rule has been processed. When this occurs, only the two most significant bits (MSB) of the five-bit field are used. The MSB selects whether the action value comes from a fixed memory location or from an I/O latch. The next MSB specifies the output mode, immediate or accumulate.

The code (000) indicates Last Fuzzy variable of Last Rule. The two most significant bits are used as described in the paragraph above. In addition, the two bits above the Type Select field are used to select the output.

The second byte always occurs on an odd address, and contains the Center and Width address index fields if the previous byte specified a membership function type and fixed center and width value. If either the center or the width were specified as floating, then their respective nibble in the odd byte is used to select the input or output.

When the first byte's Type is Last fuzzy variable or Last Fuzzy variable of Last Rule and the action is fixed, the second byte contains the action value. If action is floating, then the odd byte selects the input or output that provides the action value.

312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

e 5. Command Byte (Even Addresses)

6	5	4	3	2	1	0	
CF	I/O CONT	I/O SELECT		TYPE			2-7
MODE				TYPE 1			
MODE		OUTPUT SELECT		TYPE		0	

210

000	Last Term of Last Rule of given output
001	Last Term of Current Rule
010	MF, Symmetrical, Inclusive
011	MF, Symmetrical, Exclusive
100	MF, Left, Inclusive
101	MF, Left, Exclusive
110	MF, Right, Inclusive
111	MF, Right, Exclusive

lect **43**

00	I/O Port 0 as Input
01	I/O Port 1 as Input
10	I/O Port 2 as Input
11	I/O Port 3 as Input

ntrol **5**

0	Select from Inputs
1	Select from Outputs

6

0	Immediate, Output equals ACTION
1	Accumulate, Output equals current output plus two's complement action (-128 to +127)

7

0	Select Action from Select Byte (fixed)
1	Select Action from I/O via Select Byte (float)

t Select **43**

00	ACTION from current RULE set to Output 0
01	ACTION from current RULE set to Output 1
10	ACTION from current RULE set to Output 2
11	ACTION from current RULE set to Output 3

6

0	Select Center from memory via Select Byte (fixed)
1	Select Center from I/O via Select Byte (float)

7

0	Select Width from Memory via Select Byte (fixed)
1	Select Width from I/O via Select Byte (float)

Table 6. Select Byte (Odd Addresses)

7	6	5	4	3	2	1	0	
CENTER SELECT				WIDTH SELECT				TYPE=2-7,CF or WF=0 (FIXED)
	I/O CONT	I/O SELECT CENTER			I/O CONT	I/O SELECT WIDTH		TYPE=2-7, CF or WF=1 (FLOAT)
ACTION								TYPE=0-1, AF=0 (FIXED)
					I/O CONT	I/O SELECT ACTION		TYPE=0-1, AF=0 (FLOAT)

Key:

Width Select	(3:0)	Used as Address Index (E0-EF) for Fixed 6-bit WIDTH Value when Type =2-7 and WF=0
Center Select	(7:4)	Used as Address Index (F0-FF) for Fixed 8-CENTER Value when Type =2-7 and CF=0

I/O Select Width — **10**

00	I/O Port 0 as Width (Type=2-7 and WF=1)
01	I/O Port 1 as Width (Type=2-7 and WF=1)
10	I/O Port 2 as Width (Type=2-7 and WF=1)
11	I/O Port 3 as Width (Type=2-7 and WF=1)

I/O Control — **2**

0	Select from Inputs (Type=2-7 and WF=1)
1	Select from Outputs (Type=2-7 and WF=1)

I/O Select Center — **54**

00	I/O Port 0 as Input (Type=2-7 and WF=1)
01	I/O Port 1 as Input (Type=2-7 and CF=1)
10	I/O Port 2 as Input (Type -2-7 and CF=1)
11	I/O Port 3 as Input (Type=2-7 and CF=1)

I/O Control — **6**

0	Select from Inputs (Type=2-7 and CF=1)
1	Select from Outputs (Type=2-7 and CF=1)

ACTION — **7-0**

8-bit Action value to be applied to an output due to a winning Last Term of a Rule (TYPE=1) or Last Term of Last Rule of a given Output (Type=0), and AF=0 (Fixed)

I/O Select Action — **10**

00	I/O Port 0 as Action (Type=1-0 and AF=1)
01	I/O Port 1 as Action (Type=1-0 and AF=1)
10	I/O Port 2 as Action (Type=1-0 and AF=1)
11	I/O Port 3 as Action (Type=1-0 and AF=1)

I/O Control — **2**

0	Select from Inputs (Type=1-0 and AF=1)
1	Select from Outputs (Type=1-0 and AF=1)

312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

iming

gure 12 illustrates timing for the AL220. The three chitectural blocks that impact timing include the ltiplexed input A/D converter, the Fuzzy Processor, d the multiplexed output D/A converter.

ocessing speed is a function of both the clock rate and number of clocks (1024) required to complete data mpling and processing cycles. The clock maximum e is 10 MHz and the minimum is 1mhz.

perating Timing

set When the RESET pin is active, all the latches are ared, the digital outputs are logic low, and the analog puts hold at the level they were at prior to the ertion of reset. If RESET is active for one hundred cks or more, the analog inputs will be zero when mpling resumes. If RESET is active for less than one ndred clocks, there may be some residual of the last mpled data still present on the analog inputs when mpling resumes. When RESET is deactivated the vice begins sampling inputs during the first 1024 ck cycle.

ut Conversion Input analog values are converted to ital data and latched internally in successive periods 256 clocks each. A total of 1024 clock cycles are uired to convert all four inputs after which the version process repeats. At the maximum clock, the mple rate for each input is 10 Khz, or 100 croseconds.

cessor Timing The first 1024 clock processing cycle ins after the first input conversion cycle has com- ted. Processing cycles consist of 1024 clock cycles re- dless of the number of fuzzy variables and rules used.

zy variable and rule evaluations require four clocks h. For example, a rule with two fuzzy variables would uire 12 clocks to process. During a processing cycle er a fuzzy variable or rule is being processed each r-clock period, except for a 64-clock latency period at end of the processing cycle.

ternal Loopback Delay

en data in the output latch is internally looped back as uts, they lag behind the analog inputs by the 1024 ks of the initial sampling cycle. After that, as the ut latches are updated during processing the data lback is used as inputs.

Prescaled Operation

The device contains a loadable prescale counter that allows it to be inactive for periods of time. The feature is used to vary the rates of sampling and processing. The last location in the memory, which normally stores fixed membership function width data, may instead store a value to be loaded into the counter. The PRESCALE pin selects normal or prescaled operation.

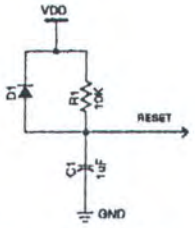
In prescale mode, the processor is inactive for periods of 1024 clocks after which the counter is incremented. When the counter reaches a value of FF, the processor is activated for a single 1024 clock period to perform Fuzzy computations and the counter is loaded again.

The Prescale pin can be connected to the Ready pin when prescale operation is desired.

Powerup Reset and Clock

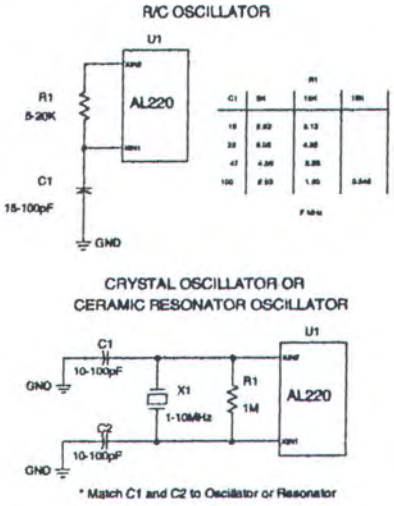
Figure 11 shows a typical power up reset circuit.

Figure 11. Powerup Reset Circuit



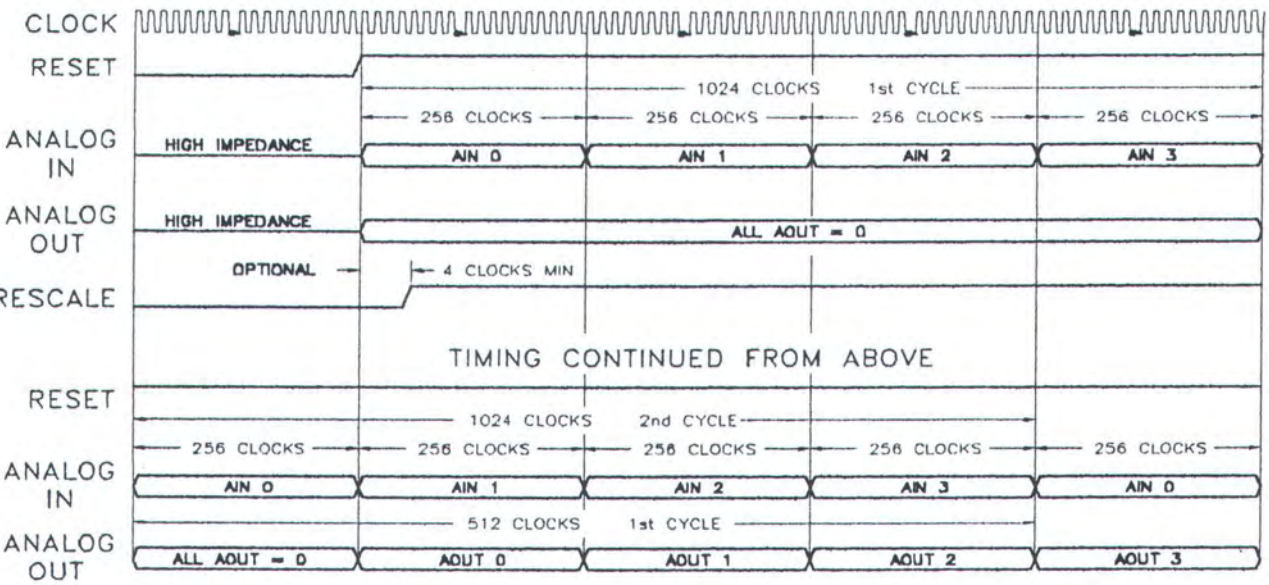
The Clock can be derived from Figure 12.

Figure 12. Clock Circuits

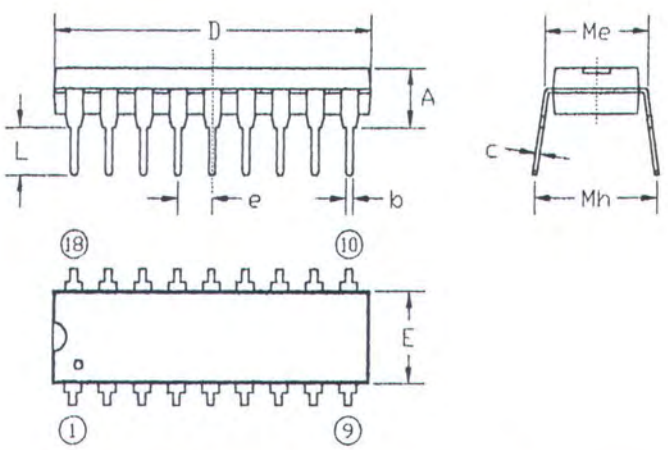


312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

ure 13. Timing Considerations



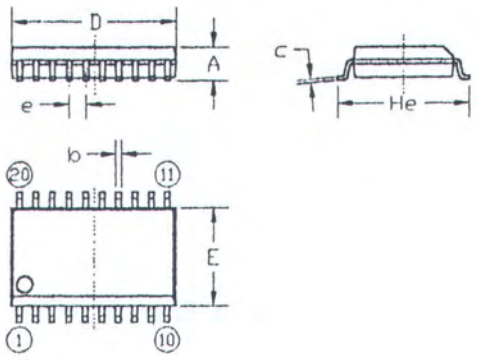
ure 14. DIP Package Mechanical Details



IT	A	b	c	D	E	e	L	Me	Mh
	4.06	1.63	0.36	23.50	6.48		3.51	8.13	10.03
n						2.54			
		1.14	0.25	23.24	6.22		3.05	7.62	7.62
	0.160	0.064	0.014	0.925	0.255		0.138	0.32	0.295
es						0.100			
		0.045	0.010	0.915	0.245		0.120	0.30	0.300

312 W. First Street, Suite 201 Sanford, FL 32771 Phone: 407-322-5608 / Fax: 407-322-5609
E-mail: info@BasiConcepts.com Web Site: www.BasiConcepts.com

ure 15. SOIC Package Mechanical Details

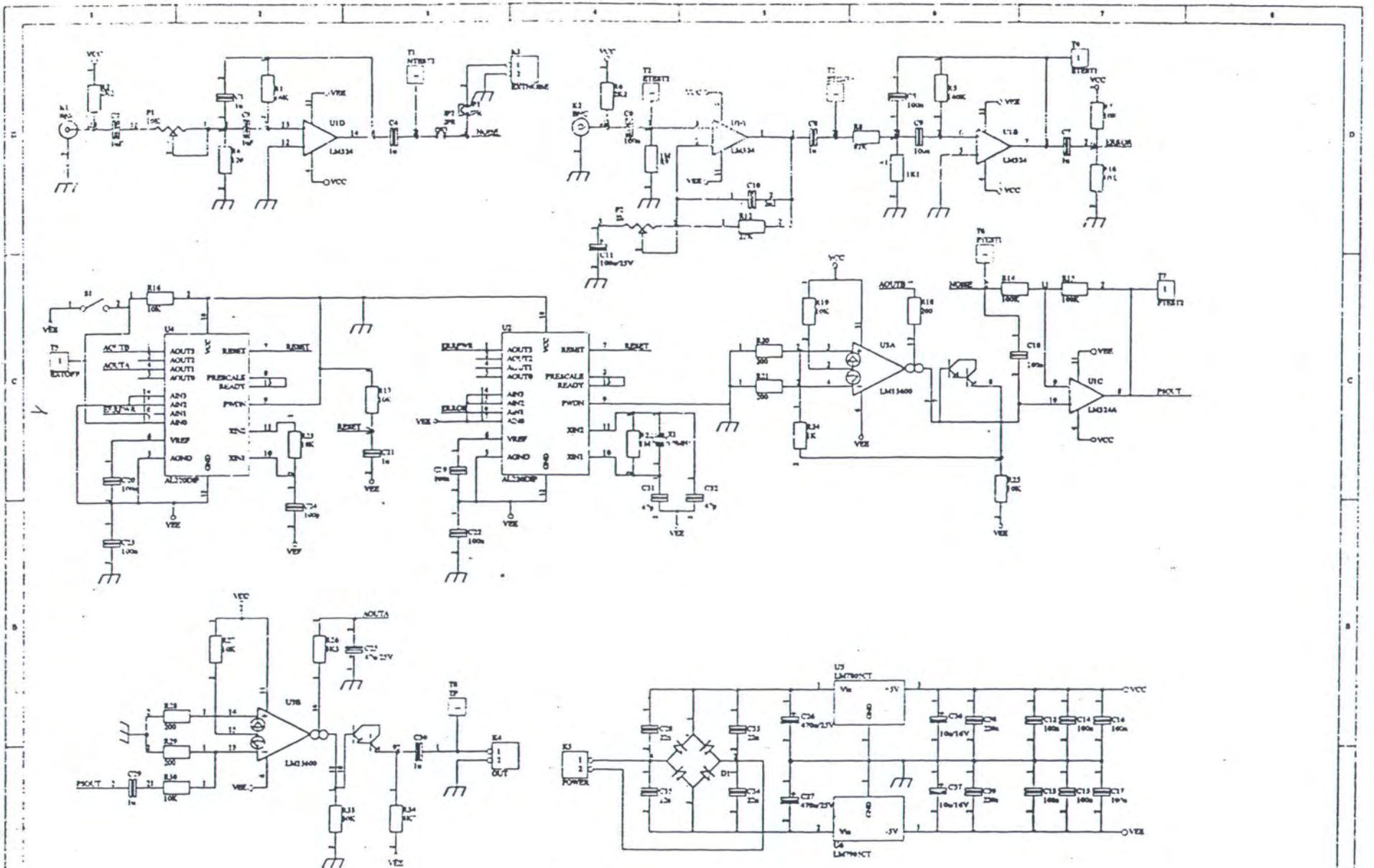


NIT	A	b	c	D	E	e	He
	2.65	0.49	0.32	13.0	7.6		10.65
m						1.27	
		0.36	0.23	12.6	7.4		10.00
	0.10	0.019	0.013	0.51	0.30		0.42
hes						0.050	
		0.014	0.009	0.49	0.29		0.39

siConcepts

W. First St. Suite 201
ford, FL 32771
-322-5608
: 407-322-5609

ted in U.S.A.



RIWAYAT HIDUP

ERWIN DIAN SANTOSO dilahirkan di Sidoarjo pada tanggal 8 Oktober 1972. Putera pertama dari tiga bersaudara dari:

Ayah : Gianto Poedjiarto

Ibu : Go Hwie Gie

Yang bertempat tinggal di Girilaya 3/34 Surabaya.

Terdaftar sebagai mahasiswa Jurusan Teknik Elektro, Fakultas Teknologi Industri Institut Teknologi Industri Sepuluh Nopember pada tahun 1991 dengan nomor registrasi pokok 2291.100.038.

Pendidikan yang telah ditempuh:

1. TK Taruna Nusa Harapan, Mojokerto, tahun 1979-1980.
2. SD Taruna Nusa Harapan, Mojokerto, tahun 1980-1986.
3. SMP Taruna Nusa Harapan, Mojokerto, tahun 1986-1989.
4. SMA Kristen Petra 2, Surabaya, tahun 1989-1991.
5. Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya, tahun 1991 sampai sekarang.